Proceedings of the 7th International Cryptology and Information Security Conference 2020

9th – 10th June 2020
Malaysia

Published by
Institute for Mathematical Research (INSPEM)
Universiti Putra Malaysia
43400 UPM Serdang
Selangor Darul Ehsan

First Print 2020

ISSN 2716-6783

FOREWORD

First and foremost, I would like to thank the Malaysian Society for Cryptology Research (MSCR) in collaboration with CyberSecurity Malaysia together with Universiti Putra Malaysia (UPM) for their continuous efforts and commitment to host this premier event, the International Cryptology and Information Security Conference for the seventh time. This biannual conference series which started in 2008 has been organized and hosted at several locations in Malaysia, beginning from Kuala Lumpur to Melaka, and then on to Langkawi, Putrajaya, Kota Kinabalu and Port Dickson in 2018. This year, however, due to unforeseen COVID-19 outbreak that is happening around the world, we gather in a digital classroom for the first time to attend this event.

CRYPTOLOGY2020 is the seventh among a series of open forum conferences for avid researchers of theoretical foundations, applications and any related issues in cryptology, information security and other underlying technologies to contribute to this body of knowledge. For this year's CRYPTOLOGY2020, participation of researchers from various disciplines is impressive and this signifies the interdisciplinary nature of the topic of the conference. A total of 13 research results are scheduled to be delivered in this forum. I hope that through this kind of events and activities, we can promote new research interests and eventually developed more expertise in this field.

Cryptology is the last line of defence in protecting information. In the absence of cryptographic measures protecting one's critical information, it cannot be ascertained that information is secured from adversaries. Hence, Malaysia must be prepared in protecting her Critical National Information Infrastructure (CNII) in the coming years as criminals and other adversaries will gain access to new technology and skills to obtain this critical information. In view of the importance of cryptography in national cyber security, National Cryptography Policy (NCP) or *Dasar Kriptografi Negara* was established under the purview of National Security Council. It has seven strategic thrusts that focus on the aspect of competency and self-reliant in cryptography towards ensuring the protection of national security, citizens' privacy and safety; and making cryptography industry as a contributor to the nation's wealth creation. CyberSecurity Malaysia together with National Security Council are the joint secretariat to monitor the implementation of the policy.

Universiti Putra Malaysia (UPM), Universiti Sains Malaysia (USM), Multimedia University (MMU), Universiti Teknikal Malaysia Melaka (UTeM), Universiti Tunku Abdul Rahman (UTAR) and several other renowned universities have been in collaboration with CyberSecurity Malaysia and the Malaysian Society for Cryptology Research for about 15 years now. I believe that these universities somehow or rather, have to a certain extent, provided a platform for R&D in this area. Other than R&D, it is also important for the experts in cryptology and researchers working in this field, to work hand in hand and enhance their networking and communications. Therefore, this conference provides a good platform for information sharing, and to showcase new technology in internet security.

Finally, to all the participants, I wish you every success in your future endeavour and a fruitful and productive conference.

Thank you.

DATO' TS. DR. HAJI AMIRUDIN BIN ABDUL WAHAB
Chief Executive Officer,
CyberSecurity Malaysia.

WELCOMING NOTES

I am very pleased to welcome speakers from countries across the world to the 7th International Cryptology and Information Security Conference 2020 (CRYPTOLOGY2020). It is our hope that participants will grab this opportunity and gain valuable experience either through formal or informal discussion during this intellectual meeting.

Cryptography is an area of research that has tremendous impact especially in the area of communication technology. In this respect, CRYPTOLOGY2020 will provide an avenue for participants to engage on current topics related to cryptology. It is also aimed at promoting and encouraging exchange of ideas and at the same time identifying areas of collaborative research between local and foreign researchers.

Information security has never become as important in our daily lives as we are experiencing today. We are now on the brink of experiencing cryptography and its deployment in every corner of our day to day experiences. Thus, research in this area has become extremely important that without continuous effort to conduct research in the area one would not be able to ascertain the degree of security being deployed. Therefore it is our responsibility to ensure this biennial gathering is held in a best possible manner such that pool of excellent ideas can be brought together to solve current and future problems.

In this conference, we have 13 papers scheduled to be presented encompassing various areas of cryptology such as theoretical foundations, applications, information security and other underlying technologies in this interesting mathematical field. I hope this conference will bring Malaysia further towards realizing and translating research into a good cryptography practices.

It goes without saying that a conference of this kind could not have been held without the committed efforts of various individuals and parties. I would like to take this opportunity to congratulate and thank everyone involved for their excellent work and in particular to Universiti Putra Malaysia (UPM) and CyberSecurity Malaysia for taking up the challenge of organizing this conference. I wish CRYPTOLOGY2020 will gives all participants great experience, enjoyable and meaningful moments. With that, I once again thank all speakers, presenters and participants in making this conference possible and a successful event.

Thank you.

PROF. DR. RAMLAN MAHMOD
President,
Malaysian Society for Cryptology Research

EDITORIAL PREFACE

Since the time of Julius Caesar and possibly up until the Greek era, cryptography (a word that is derived from the Greek term "cryptos") has been an integral tool for organizations (and indeed for individuals too) to ensure information that is intended only for authorized recipients remain confidential only to this set of people. Cryptography had far reaching implications for organizations in the event information leakage occurred. Often referred to as the "last bastion of defence" after all other mechanisms had been overcome by an adversary, encrypted information would still remain useless to the attacker (i.e. that is, under the usual security assumptions). Nevertheless, this simple fact has remained oblivious to the practitioners of information security omitting cryptographic mechanism for data being transferred and also during storage.

Fast forward to World War 2, the war between cryptographic and cryptanalytic techniques. While the Germans were efficiently transferring information via the Enigma encryption machine, the Allies in Bletchley Park, England were busy intercepting these ciphered information being transmitted via telegraph by the Germans. Leading mathematicians, linguists, engineers etc. were all working to cryptanalyze these ciphers in the most information way. It is here that the first electrical machine (i.e. the "bomba") was born and revolutionized computing. Post World War 2 saw the emergence of the "computer". Every organization that had to process data had to acquire a computer so as not to be left behind by their competitor. The banking sector advanced on a global scale due to the invention of the computer. Techniques to secure information among the headquarters of these banks had to be developed. Encryption procedures using the same key (i.e. symmetric encryption) played this role in the early days. Then came the unthinkable problem computers were being deployed almost everywhere. How is it possible to deploy cryptographic keys in secure manner so that symmetric encryption could take place? Thus, leading to the so-called "key distribution" problem. It was not until 1975, when Diffie and Hellman provided us with a secure key exchange method and in 1976 when Rivest, Shamir and Adleman with the "asymmetric encryption" scheme (i.e. to encrypt using key e and decrypt using key d, where $e \neq d$). Since then, cryptographic procedures evolved, not only playing the role of ensuring confidentiality of data, but also to ensure integrity and authenticity of data. It is also able to ensure that non-repudiating of data does not occur.

Mechanisms to transfer and store data has changed of the centuries and more so every 5 years (in this modern age). Cryptography that has long existed before mechanisms changed from manual telegraphic electrical electronic (WAN/LAN/internet) wired until wireless procedures, has to be properly deployed in order to maintain a high level of security confidence among the stakeholders of a certain organization. The concept of securing information via encryption procedures has to be properly understood in order to avoid a null intersection to occur between cryptography and computer security practitioners. This scenario would not be to the best interest for stakeholders. As a "friendly" reminder, this scenario could already been seen in other discipline of knowledge where the "minuting" ("minute-ting") of knowledge has forced the original body of knowledge to look as though it is independent and disassociated. Ever since mass usage of computers became a reality, computer security issues have never been this complicated. However, as the human race advances so will ingenious ideas emerge to overcome challenges.

It is hoped that CRYPTOLOGY2020 will not only provide a platform for every participant to exchange ideas in their respective fields, but also to exchange new ideas on a broader scale for the advancement of the field of cryptology and computer security. The organizing committee hopes every participant will have an enjoyable and beneficial conference.

Thank you.

Editorial Board,
CRYPTOLOGY2020

Table of Contents

General Chair	Muhammad Rezal Kamel Ariffin
International Program Committee	Abderrahmane Nitaj
	Ahmad Izani Md. Ismail
	Amr M. Youssef
	Bharathwaj Muthuswamy
	Kamel Ariffin Mohd Atan
	Kaoru Kurosawa
	Keith Martin
	Lamberto Rondoni
	Maslina Daud
	Mohamed Ridza Wahiddin
	Mohd Salmi Md Noorani
	Rennato Renner
	Sazali Sukardi
	Shahrin Sahib
	Solahuddin Shamsuddin
	Yanbin Pan
Executive Editors	Muhammad Rezal Kamel Ariffin
	Amir Hamzah Abd Ghafar
Technical Program Committee	Amir Hamzah Abd Ghafar
	Aniza Abd Ghani
	Anuar Mat Isa
	Chin Ji Jian
	Denis Wong Chee Keong
	Faridah Yunos
	Goi Bok Min

	Hailiza Kamarulhaili
	Hayder Natiq Kadhim
	Hazlin Abdul Rani
	Heng Swee Huay
	Masnida Hussin
	Moesfa Soehaila Mohamad
	Muhammad Asyraf Asbullah
	Muhammad Reza Z'aba
	Muhammad Rezal Kamel Ariffin
	Normahirah Nek Abd Rahman
	Nur Azman Abu
	Syh Yuan Tan
	Wong Tze Jin
	Yap Wun She
Committee Members	Aniza Abdul Ghani
	Faridatul Akhma Ishak
	Hazlin Abdul Rani
	Muhammad Asyraf Asbullah
	Nor Azlida Aminudin
	Nur Raidah Salim
	Nur Sumirah Mohd Dom
	Wan Zariman Omar
	Zahari Mahad
Illustration & Art Work	Zahari Mahad

Ensuring Information Security Through the Use of Cryptography

Amirudin Abdul Wahab

CyberSecurity Malaysia

ABSTRACT

The presentation will highlight the current and emerging trends of cyber threats landscape, and how the advancement technology that includes digital transformation in a new normal environment brings with it in parallel new risks with regard to information insecurity. This presentation will briefly share various cyber incidents that undermine information security in the aspects of confidentiality, integrity and availability. From here, the presentation will focus on the importance of cryptographic solutions as the last line of defence to protect information security in the aspects of confidentiality, integrity and authenticity. The presentation will also share Malaysia's initiatives in addressing relevant issues covering the triad of people, process and technology.

Security of Rabin-p Key Encapsulation Mechanism

Ji-Jian Chin[1] and **Moesfa Soeheila Mohamad**[*2]

[1]*Faculty of Engineering, Multimedia University, Cyberjaya, Selangor.*
[2]*Information Security Lab, MIMOS Berhad, Kuala Lumpur.*

E-mail: soeheila.mohamad@mimos.my
[*]*Corresponding author*

ABSTRACT

The Rabin-p key encapsulation mechanism (KEM) was proposed by Asbullah et al. in 2019 for the MySEAL New Cryptographic Algorithm (AKBA) initiative. The authors proposed a public key encryption scheme which is a variant of the Rabin cryptosystem in that the modulus is multiprime and the private key consists of only one prime, thus saving computation and storage power in terms of the private key component. However, it is known that the scheme is deterministic and not secure against chosen-plaintext attacks. Therefore the authors conducted a Dent transform to convert it into a KEM that is indistinguishably-secure against chosen ciphertext attacks in the random oracle model. However the authors did not provide a formal treatment to the security analysis, only some statements claiming to satisfy the IND-CCA2 requirements. This work provides the formal treatment for the scheme with regards to the security proof.

Keywords: key encapsulation mechanism, Rabin encryption, integer factorization

1 INTRODUCTION

In 2015, the MySEAL initiative by CyberSecurity Malaysia Berhad brought together cryptographers from all around Malaysia to provide a rigorous study on cryptographic algorithms that are deemed ‘safe’ to be deployed by the Malaysian government. The first list of trusted cryptographic algorithms chosen from standards and other nations’ recommended list, AKSA, was published in November 2017 on the MySEAL website (MySEAL, 2019). The AKSA list consists of twelve symmetric block ciphers, three symmetric stream ciphers, three digital signatures, six public key encryption schemes, two key agreement schemes, twenty hash functions and its variants, three prime number generators and nine deterministic random bit generators.

Following that, a call for proposals for new algorithms by Malaysian cryptographers was initiated. Upon receiving numerous proposals and completion of two rounds of rigorous analysis, algorithms that did not fulfil the AKBA proposal's criteria were eliminated. The outcome of the AKBA exercise yielded two remaining algorithms. The Rabin-p key encapsulation mechanism (KEM) is one of the finalists.

The Rabin-p KEM (Asbullah et al., 2019) is constructed from the Rabin-p public key encryption (PKE) scheme by Asbullah and Ariffin (2016). Its security is based on the hardness of factoring, similar to the original Rabin encryption scheme by (Rabin, 1979). However, the slight modification of Rabin-p includes changing the public key from $N = pq$ to $N = p^2q$ and using only p as the private key. This provides for accurate decryption, eliminating the decryption error of the original Rabin encryption scheme as well as shaving off q as an additional private key portion.

Whilst it is known that the Rabin-p cryptosystem does not satisfy indistinguishability, the designers of the scheme claim that the scheme satisfies one-wayness. Using this property, the designers then proceeded to reinvent the Rabin-p encryption scheme into a KEM following the transformation proposed by Dent (2003). Initially claiming the KEM to be secure against indistinguishable adaptive chosen-ciphertext attacks (IND-CCA2), the designers then downplayed the security claims to only satisfy indistinguishability under chosen-plaintext attacks (IND-CPA) due to the work of Paillier and Villar (2006). To the best of our knowledge, there exists no proof to the designers' claims that their KEM satisfies IND-CPA or IND-CCA2.

This work aims to provide such formal treatment to the Rabin-p KEM in order to fulfil the MySEAL's call for cryptanalysis. In this work, we show that the Rabin-p KEM does indeed satisfy IND-CCA2 under the random oracle model, following Dent's transformation. This is notwithstanding the claims of the designers that the Rabin-p KEM achieves only IND-CPA security due to Paillier and Villar (2006) which only show the impossibility of single private key encryption schemes (such as the Rabin cryptosystem) to achieve IND-CCA2 security. However, since the Rabin-p KEM is NOT an encryption scheme but a KEM, this result does not apply. Therefore, here we instantiate the proof from (Dent, 2003, Appendix B) to tailor to Rabin-p, showing concrete security bounds of an IND-CCA2 adversary's advantage against Rabin-p KEM.

The rest of the paper is as follows: We begin by providing notations and a review of PKEs and KEMs in Section 2, then review the Rabin-p PKE and KEM in Section 3. The main contribution of this work can be found in Section 4 where we provide the proof of security for the Rabin-p KEM. We also share some insight on recommended key lengths for Rabin-p KEM in order to achieve similar security level to that of 128-bit AES in Section 5. Finally we conclude in Section 6.

2 PRELIMINARIES

In this section we provide some preliminaries for notations and cryptographic primitives.

We denote $\{0, 1\}^*$ as the set of all bit strings and $\mathbb{Z}_p$ as the set of positive integers modulo p, where p is a large prime number. The notation $a \xleftarrow{\$} S$ denotes sampling a random element

a uniformly from a finite set S, therefore $x \xleftarrow{\$} \{0,1\}^n$ shows randomly sampling a bitstring of length n whereas $b \xleftarrow{\$} \mathbb{Z}_p$ shows randomly sampling an integer b from the set of $\mathbb{Z}_p$.

We denote a function $\mathsf{negl}(n)$ as negligible if for all polynomials p there is a constant N_p where for any $n \geq N_\mathsf{p}$, $\mathsf{negl}(n) \leq \frac{1}{\mathsf{p}(n)}$.

2.1 Public Key Encryption (PKE)

Let $\mathcal{M}_E$ be the message space and $\mathcal{C}_E$ be the ciphertext space of a public key encryption (PKE) scheme E. A PKE scheme E consists of three algorithms:

1. $E.\mathsf{KGen}(1^n) \rightarrow (pk, sk)$: The key generation algorithm that takes in the security parameter and outputs a public/private key pair.
2. $E.\mathsf{Enc}(m, pk) \rightarrow C$: the encrypt function that takes in a user's public key pk and a message $m \in \mathcal{M}_E$ and outputs a ciphertext $C \in \mathcal{C}_E$.
3. $E.\mathsf{Dec}(C, sk) \rightarrow m$: the decrypt function that takes in a user's corresponding private key sk and a ciphertext $C \in \mathcal{C}_E$ and recovers the message $m \in \mathcal{M}_E$.

It is required for correctness that $E.\mathsf{Dec}(E.\mathsf{Enc}(pk, m), sk) = m$.

The security game for E against one-way chosen plaintext attacks (OW-CPA) is defined as the advantage of adversary $\mathcal{A}$ winning the following OW-CPA experiment, shown in Figure 1.

```
$Exp_{E,\mathcal{A}}^{OW-CPA}(1^n)$
$(pk, sk) \xleftarrow{\$} \mathsf{KGen}(1^n)$
$(state) \xleftarrow{\$} \mathcal{A}(1^n, pk)$
$m \xleftarrow{\$} \mathcal{M}$
$c \xleftarrow{\$} \mathsf{Enc}(pk, m)$
$m' \xleftarrow{\$} \mathcal{A}(1^n, pk, c, state)$
if $m^* = m'$ then
    return $1$
 else
    return $0$
```

Figure 1: OW-CPA experiment against E.

The security of E is then defined as advantage of the adversary $\mathcal{A}$ as follows:

$$Adv_{E,\mathcal{A}}^{OW-CPA}(1^n) = \Pr\left[Exp_{E,\mathcal{A}}^{OW-CPA}(1^n) = 1\right]$$

2.2 Key Encapsulation Mechanism (KEM)

Let $\mathcal{K}_{KEM}$ be the key space and $\mathcal{C}_{KEM}$ be the ciphertext space. A key encapsulation mechanism KEM consists of three algorithms:

1. $KEM.\mathsf{KGen}(1^n) \rightarrow (pk, sk)$: The key generation algorithm that takes in the security parameter and outputs a public-private key pair.
2. $KEM.\mathsf{Encap}(pk) \rightarrow (K, C)$: the key encapsulation function that takes in a user's public key pk and outputs a ciphertext $C \in \mathcal{C}_{KEM}$ and a key $K \in \mathcal{K}_{KEM}$ using its random coins.
3. $KEM.\mathsf{Decap}(C, sk) \rightarrow K$: the decrypt function that takes in a user's corresponding private key sk and a ciphertext $C \in \mathcal{C}_{KEM}$ and recovers the key $K \in \mathcal{K}_{KEM}$.

The security game for KEM against indistinguishable chosen ciphertext attacks (IND-CCA2) is defined as the advantage of adversary $\mathcal{B}$ winning the following IND-CCA2 experiment, shown in Figure 2. The security of KEM is then defined as advantage of the adversary $\mathcal{B}$ as follows:

$$Adv^{\text{IND-CCA2}}_{KEM,\mathcal{B}}(1^n) = \left| \Pr\left[Exp^{\text{IND-CCA2}}_{KEM,\mathcal{B}}(1^n) = 1 \right] - \frac{1}{2} \right|$$

```
$Exp^{\text{IND-CCA2}}_{KEM,\mathcal{B}}(1^n)$
$(pk, sk) \xleftarrow{\$} KEM.\mathsf{KGen}(1^n)$
$(\mathsf{state}) \xleftarrow{\$} \mathcal{A}^{KEM.\mathsf{Decap}(sk,.)}(1^n, pk)$
$(K_0^*, C^*) \xleftarrow{\$} KEM.\mathsf{Encap}(pk)$
$(K_1^*) \xleftarrow{\$} \mathcal{K}_{KEM}$
$b \xleftarrow{\$} \{0,1\}$
$b' \xleftarrow{\$} \mathcal{A}^{KEM.\mathsf{Decap}(sk,.)}(1^n, pk, K_b^*, C^*, \mathsf{state})$
if $b = b'$ then
    return $1$
 else
    return $0$
 fi
```

Figure 2: IND-CCA experiment against KEM

3 THE RABIN-P PKE AND KEM

In this section we review the Rabin-p public key encryption (PKE) scheme (Asbullah and Ariffin, 2016) and the derived KEM (Asbullah et al., 2019).

3.1 Rabin-p PKE

Let $msg \overset{\$}{\leftarrow} \{0,1\}^*$ and $\mathsf{Parse}(\cdot)$ be a function that maps bitstrings to elements in $\mathcal{M} = \left\{0, 2^{2n-1}\right\}$. The Rabin-p PKE scheme E consists of three algorithms as described in Figure 3.

```
$E.\mathsf{KGen}(1^n) \rightarrow (pk = N, sk = p)$
  $p, q \overset{\$}{\leftarrow} \mathbb{Z}_l$ :
      $2^n < l < 2^{(n+1)}, p, q \equiv 3(\mathrm{mod} 4)$
  $N = p^2 q$
  return $(N, p)$

$E.\mathsf{Enc}(m, pk = N) \rightarrow C$
  $m = \mathsf{Parse}(msg)$ :
      $0 < m < 2^{2n-1}$and$gcd(m, N) = 1$
  $C = m^2(\mathrm{mod} N)$
  return $(C)$

$E.\mathsf{Dec}(C, sk = p) \rightarrow m$
  $w \equiv C(\mathrm{mod} p)$
  $m_p \equiv w^{\frac{p+1}{4}}(\mathrm{mod} p)$
  $i = \frac{c - m_p^2}{p}$
  $j \equiv \frac{i}{2m_p}(\mathrm{mod} p)$
  $m_1 = m_p + jp$
  if $m_1 < 2^{2n-1}$ then
      return $m = m_1$
  else
      return $m = p^2 - m_1$
```

Figure 3: Rabin-p PKE

```
$KEM.\mathsf{KGen}(1^n) \rightarrow (pk = N, sk = p)$
  $p, q \overset{\$}{\leftarrow} \mathbb{Z}_l : 2^n < l < 2^{(n+1)}$,
                  $p, q \equiv 3(\mathrm{mod} 4)$
  $N = p^2 q$
  Select $KDF : \mathbb{Z}_{2^{2n-1}} \rightarrow \mathcal{K}$
  Select $H : \mathbb{Z}_{2^{2n-1}} \rightarrow \mathcal{C}$
  return $(N, p, KDF, H)$

$KEM.\mathsf{Encap}(pk = N) \rightarrow (K, C)$
  $x \overset{\$}{\leftarrow} \mathbb{Z}_l : 2^{3n/2} < l < 2^{2n-1}$
  $C_1 = x^2(\mathrm{mod} N)$
  $C_2 = H(x)$
  $C = (C_1, C_2)$
  $K = KDF(x)$
  return $(K, C)$

$KEM.\mathsf{Decap}(C, sk = p) \rightarrow K$
  Parse $C = (C_1, C_2)$
  $w \equiv C_1(\mathrm{mod} p)$
  $x_p \equiv w^{\frac{p+1}{4}}(\mathrm{mod} p)$
  $i = \frac{C_1 - x_p^2}{p}$
  $j \equiv \frac{i}{2x_p}(\mathrm{mod} p)$
  $x_1 = x_p + jp$
  if $x_1 < 2^{2n-1}$ then
      $x = x_1$
  else
      $x = p^2 - x_1$
  if $C_2 \neq H(x)$return $\perp$
  $K = KDF(x)$
  return $K$
```

Figure 4: Rabin-p KEM

3.2 Rabin-p KEM

Let $\mathcal{K} = \{0,1\}^{keylen}$ and $\mathcal{C}_{KEM} = \{0, N\}$. Furthermore, define KDF to be a pseudorandom function and H to be a hash function. The Rabin-p KEM consists of three algorithms as described in Figure 4.

4 SECURITY ANALYSIS

We provide the IND-CCA2 proof for the Rabin-p KEM in this section.

Theorem 4.1. *Given a OW-CPA secure Rabin-p PKE scheme, a pseudorandom key derivation function KDF and a hash function H, the Rabin-p KEM is secure against IND-CCA2 attacks with the following advantage:*

$$Adv_{KEM,\mathcal{B}}^{IND-CCA2}(n) \leq Adv_{PKE,\mathcal{A}}^{OW-CPA}(n) + \frac{q_D}{2^{Hashlen}} + \frac{q_D}{2^{2n-1}}$$

where q_D is the number of decapsulation queries made by $\mathcal{A}$ and $Hashlen$ is the length of the output of H.

Proof. We model the security of IND-CCA2 Rabin-p KEM as a game where $\mathcal{A}$ breaks the OW-CPA Rabin-p PKE scheme using an adversary $\mathcal{B}$ that breaks IND-CCA2 of Rabin-p KEM. During initiation, $\mathcal{A}$ receives the public key $pk = N = p^2q$ and a challenge ciphertext C^* of which it must invert (i.e. produce x^* such that $C^* = (x^*)^2(\text{mod} N)$ using the help of $\mathcal{B}$.

$\mathcal{A}$ maintains two lists for its KDF and H oracles, $KDF - list$ and $H - list$ respectively. $\mathcal{A}$ passes $pk = N$ to $\mathcal{B}$, stores C^* aside for the challenge phase, and simulates KDF and H as random oracles. Upon each KDF or H query, on input of x_i from $\mathcal{B}$, $\mathcal{A}$ checks if $C_1^* = E.\mathsf{Enc}(x_i, pk)$. If true, $\mathcal{A}$ ends the game and returns x_i as the solution m^* to the challenge C^*. Otherwise, $\mathcal{A}$ provides the following oracles for $\mathcal{B}$ to query adaptively:

1. KDF queries: $\mathcal{A}$ checks if $(x_i, K_i) \in \{KDF - list\}$. If the entry is not found, $\mathcal{A}$ samples $K_i \xleftarrow{\$} \mathcal{K}_{KEM}$, stores $(x_i, K_i) \in \{KDF - list\}$ and returns K_i to $\mathcal{B}$.
2. H queries: $\mathcal{A}$ checks if $(x_i, H(x_i)) \in \{H - list\}$. If the entry is not found, $\mathcal{A}$ samples $H(x_i) \xleftarrow{\$} \{0,1\}^{Hashlen}$, stores $(x_i, H(x_i)) \in \{H - list\}$ and returns $H(x_i)$ to $\mathcal{B}$.
3. Decap queries: On input of $(C_i = (C_{(1,i)}, C_{(2,i)}))$ from $\mathcal{B}$, one of the following two scenarios will cause $\mathcal{A}$ to abort the game:

   (a) if $C_{(1,i)} = C^*$.

   (b) if $(x_i, C_{(2,i)}) \in \{H - list\}$ such that $C^* = E.\mathsf{Enc}(x_i, pk)$.

   Otherwise, $\mathcal{A}$ generates or retrieves the corresponding x_i to $C_{(2,i)}$ from $\{H - list\}$, generates or retrieves K_i from $(x_i, K_i) \in \{KDF - list\}$ and returns K_i to $\mathcal{B}$.

Once $\mathcal{B}$ completes the training phase and outputs a state to be challenged on, $\mathcal{A}$ produces $K_0^* \xleftarrow{\$} KEM.\mathsf{Encap}(pk)$ and $K_1^* \xleftarrow{\$} \mathcal{K}_{KEM}$. Next, $\mathcal{A}$ flips a bit $b \xleftarrow{\$} \{0,1\}$ and passes K_b^* and $C_{KEM}^* = (C^*, C_2^*)$ to $\mathcal{B}$. After receiving this challenge, $\mathcal{B}$ can continue querying oracles with the exception of decapsulation query on C^*. Finally, $\mathcal{B}$ must output a guess b'.

If $\mathcal{A}$ has not ended the game at this point, it then samples $x^* \xleftarrow{\$} \left\{0, 2^{2n-1}\right\}$ and outputs x^* as its solution.

It remains to calculate the probability of $\mathcal{B}$ running to completion and the abort scenarios.

The game ends when $\mathcal{A}$ wins. This corresponds to the advantage of $\mathcal{A}$: $Adv_{PKE,\mathcal{A}}^{OW-CPA}(n)$. This happens on the event that $C_1^* = E.\mathsf{Enc}(x_i, pk)$ occurs during KDF and H queries. $\mathcal{A}$ wins when it returns x_i as the solution to the challenge ciphertext C^*.

The game also ends following two scenarios that happen during decapsulation queries that cause $\mathcal{A}$ to abort. This is reviewed below together with their corresponding probabilities:

1. if $(C_{(1,i)}) = C^*$, this means $\mathcal{B}$ issued a decapsulation query on the challenge ciphertext. This might happen before $\mathcal{B}$ wishes to switch to challenge phase and happens with an upper-bound probability of $\frac{1}{2^{2n-1}}$ as a random x_i is sampled each time from $\mathbb{Z}_l$ where $2^{3n/2} < l < 2^{2n-1}$. With q_D queries, the probability of this happening throughout the game is $\frac{q_D}{2^{2n-1}}$.
2. if $(x_i, C_{2,i}) \in \{H - list\}$ such that $C^* = E.\mathsf{Enc}(x_i, pk)$, $\mathcal{B}$ has caused a collision in the random oracle query. This happens with probability $\frac{1}{2^{HashLen}}$. With q_D queries, the probability of this happening throughout the game is $\frac{q_D}{2^{HashLen}}$.

Putting them together, the chances of $\mathcal{B}$ running to completion and winning the game is given as in Theorem 4.1:

$$Adv_{KEM,\mathcal{B}}^{IND-CCA2}(n) \leq Adv_{PKE,\mathcal{A}}^{OW-CPA}(n) + \frac{q_D}{2^{Hashlen}} + \frac{q_D}{2^{2n-1}}$$

□

However, we do note a few points of contention that may raise further concerns. The first issue is that the existence of a mapping protocol $parse$ that maps from bitstrings of variable length to elements in $\mathcal{M}_E = \left\{0, 2^{2n-1}\right\}$ seems like folklore. However, this very function is used to map inputs for the Rabin-p PKE's encryption algorithm from bitstrings to integers with the condition that $\gcd(m, N) = 1$. The designers make no note of what will happen when $\gcd(m, N) \neq 1$, whether the message will be remapped, or the encryption simply aborts. This additional control may potentially leak information to an adversary.

Secondly this proof does not take into account the generation of prime numbers, nor the range of safe primes within the encapsulation algorithm. That analysis is beyond the scope of this paper. Although the authors did provide some ad hoc analysis of the Rabin-p PKE with regards to attack vectors from Coppersmith, Novak, and other mathematical analysis, it remains uncertain whether whether the exhaustive list of algebraic attacks is made known.

5 RECOMMENDED KEY LENGTHS

Since the proof of security is tight and the added advantage of the IND-CCA2 adversary is only linear to the advantage of the OW-CPA adversary, we affirm that the keylength of 3072-bits for the modulus N is sufficient to provide security at 128-bit AES security level. Table 1 lists the recommended security parameter lengths for Rabin-p KEM.

This is done by instantiating the advantage equation from Theorem 4.1 to the security parameter of $k = 1024$ corresponding to the prime number size, selecting SHA3-512 as the hash function with $Hashlen = 512$, and bounding decapsulation queries $q_D = 2^{30}$ following Coron's example Coron (2000). Thus we have:

$$\begin{aligned} Adv_{KEM,\mathcal{A}}^{IND-CCA2}(n) &\leq Adv_{PKE,\mathcal{A}}^{OW-CPA}(n) + \frac{q_D}{2^{Hashlen}} + \frac{q_D}{|\mathcal{M}|} \\ &\leq Adv_{PKE,\mathcal{A}}^{OW-CPA}(n) + 2^{30-512} + 2^{30-1024} \\ &\leq Adv_{PKE,\mathcal{A}}^{OW-CPA}(n) + 2^{-482} + 2^{-994} \end{aligned}$$

Since the addition of the terms from the decapsulation queries are linear, if $Adv_{PKE,\mathcal{A}}^{OW-CPA}(n)$ is a negligible function $\mathsf{negl}(n)$ then $Adv_{KEM,\mathcal{A}}^{IND-CCA2}(n)$ remains $\mathsf{negl}(n)$. Hence, NIST guidelines can still be followed to assume 3076-bits for factoring to provide the equivalence to 128-bit security as published on keylength.com (BlueKrypt, 2019).

Security Level	Modulus Size (bits)	Prime Size (bits)
128	3072	1024
192	7608	2560
256	15360	5120

Table 1: Rabin-p KEM recommended modulus length for 2016-2030 & beyond

6 CONCLUSION

In this work, we have shown that the Rabin-p KEM is IND-CCA2 secure assuming the Rabin-p PKE achieves OW-CPA with more concrete bounds in regards to the number of decapsulation queries, hash length and KEM message space. We also affirm that the proposed key lengths to achieve equivalent 128-bit AES security is sufficient.

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Education of Malaysia in providing financial support for this work through the Fundamental Research Grant Scheme (FRGS/1/2019/ICT04/MMU/02/5).

REFERENCES

Asbullah, M. and Ariffin, M. (2016). Design of Rabin-like cryptosystem without decryption failure. *MJMS*, 10(S):1–18.

Asbullah, M. A., Ariffin, M. R. K., and Mahad, Z. (2019). Rabin-p encapsulation mechanism. AKBA MySEAL, Rabin-p KEM Proposal. `https://myseal.cybersecurity.my/en/akba.html`.

BlueKrypt (2019). Cryptographic key length recommendation. `https://www.keylength.com/`.

Coron, J. (2000). On the exact security of full domain hash. In Bellare, M., editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, Heidelberg.

Dent, A. (2003). A designer's guide to KEMs. In Paterson, K., editor, *IMA International Conference on Cryptography and Coding 2003*, volume 2898 of *LNCS*, pages 133–151. Springer, Heidelberg.

MySEAL (2019). MySEAL homepage. `https://myseal.cybersecurity.my/en/index.html`.

Paillier, P. and Villar, J. L. (2006). Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In Lai, X. and Chen, K., editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 252–266. Springer, Heidelberg.

Rabin, M. (1979). Digitalized signatures and public-key functions as intractable as factorization. Computing Science Technical Report TR-212, MIT Laboratory for Computer Science.

Cryptanalysis of RSA Cryptosystem via the Continued Midpoint Subdivision Analysis

Muhammad Rezal Kamel Ariffin[*1,2], **Wan Nur Aqlili Ruzai**[1], **Muhammad Asyraf Asbullah**[1,3], and **Zahari Mahad**[1]

[1]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400, Selangor, Malaysia*
[2]*Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400, Selangor, Malaysia*
[3]*Centre of Foundation Studies for Agriculture Science, Universiti Putra Malaysia, 43400, Selangor, Malaysia*

E-mail: rezal@upm.edu.my, wannuraqlili@gmail.com, ma_asyraf@upm.edu.my, zaharimahad@upm.edu.my
[*]*Corresponding author*

ABSTRACT

In RSA cryptosystem, let $N = pq$ be its modulus and e be its public exponent. Throughout the years, numerous algebraic cryptanalysis conducted upon RSA tackle on the mathematical structure of the key equation $ed - k\phi(N) = 1$ where $\phi(N) = (p-1)(q-1)$ is the Euler's totient function. In this paper, we perform an attack via the continuos midpoint subdvision analysis upon an interval containing $(p-1)(q-1)$ along with continued fractions of certain related numbers. Thus, based on such analysis, we exponentially raise the security boundary of private exponent d as opposed to the previous results with similar approach.

Keywords: RSA cryptosystem, integer factorization problem, algebraic cryptanalysis, continued fractions

1 INTRODUCTION

The concept of using distinct keys for encryption and decryption process between two communicating parties emerged in 1976 within the seminal work of Diffie and Hellman (1976). However, the implementation of asymmetric cryptography is not a popular option among the practitioners until the introduction of the RSA cryptosystem in 1978. The acronym of RSA was primarily due to the names of its inventors; Rivest, Shamir and Adleman (Rivest et al., 1978). This cryptosystem is being implemented in digital world with the aims of providing privacy, authenticity and security of data.

RSA is comprised of these mathematical operations. Suppose the public modulus of RSA is given by $N = pq$ which represents the product of two strong unknown primes p and q. In key generation algorithm, the positive integers e and d are related by the modular relation $ed \equiv 1(\text{mod } \phi(N))$ where $\phi(N) = (p-1)(q-1)$ is the Euler's totient function. The relation between both public and private exponents e and d also can be expressed in key equation form given by $ed - k\phi(N) = 1$ for a positive integer k. The algorithm outputs the public key tuple (N, e) and kept the private key tuple (p, q, d). In encryption algorithm, on input of the message M and public exponent e, one easily computes $C \equiv M^e \pmod N$ for ciphertext C. In the reverse process, one simply computes $M \equiv C^d \pmod N$ to decrypt the message M from the given legitimate C and private exponent d.

Essentially, one of the security feature of RSA depends on the hardness of finding the prime factors p and q given the large integer N. However, most successful attacks proposed on RSA exploited the algebraic cryptanalysis techniques regardless directly targeted onto this well known factoring problem. As an example, the notable work by Wiener (1990) proved that the secret parameters k and d can be computed efficiently using the continued fractions algorithm if $d < \frac{1}{3}N^{0.25}$. Later, Boneh and Durfee (2000) proposed that RSA can be broken by Coppersmith's lattice reduction-based method if the decryption exponent $d < N^{0.292}$. Inspired by Wiener's attack on RSA, Bunder and Tonien (2017) proved that RSA is insecure if $d < \sqrt{\frac{8N^{1.5}}{e}}$ via the continued fractions expansion of $\frac{e}{N'}$. Later, Tonien (2018) extends the work of Bunder and Tonien (2017) and showed that RSA is vulnerable if $d < \sqrt{\frac{8tN^{1.5}}{e}}$ for an arbitrary parameter t with time complexity $\mathcal{O}(t \log N)$; also via the continued fractions method.

Our contributions. In this work, we extend our proposed method that is the continuous midpoint subdivision analysis on the interval containing the Euler's totient function (i.e. $\phi(N) = (p-1)(q-1)$) of an RSA cryptosystem. We prove that the unknowns parameters d and k can be found among the convergents of the continued fractions expansion of $\frac{e}{\lfloor \mu_{(i,j)} \rfloor}$ given by $\mu_{(i,j)} = N + 1 - \sqrt{N}\left(\frac{1+2j}{2^i} + \frac{3(2^{i+1})-3-6j}{2^{i+2}}\sqrt{2}\right)$ whenever $d < \sqrt{\frac{2^i \cdot B}{e \cdot (A + 2^{i+2})}}$ where $A = (\frac{3}{\sqrt{2}} - 2)\sqrt{N}$ and $B = N^2 - \frac{6}{\sqrt{2}}N^{1.5} + \frac{9}{2}N$. Afterwards, we solve for the prime factors of modulus $N = pq$. At the end of this work, we show that we improve the upper cryptanalytic bound of private exponent d as opposed to the previous results with similar approach (i.e. via the continued fractions algorithm).

Paper organization. The layout of the paper is structured as follows. We begin by highlighting some significant existing results and introducing our proposed method in Section 2. We present our analysis and discussion; also we include the relevant example in Section 3. We briefly summed up our work in Section 4.

2 PRELIMINARIES

In this section, we provide some previous results related to cryptanalysis on RSA via the continued fractions that will be used thoroughly in this paper.

Definition 2.1. (Continued Fractions.) *(Hardy and Wright, 1965) The continued fractions expansion of a real number X is an expression having the form*

$$X = [x_0, x_1, x_2, \cdots] = x_0 + \cfrac{1}{x_1 + \frac{1}{x_2 + \frac{1}{x_3 + \cdots}}}$$

where x_0 is an integer and x_i are the positive integers such that $i > 0$.

The following theorem ensures that the unknown integers y and z can be determined from the list of convergents of continued fractions expansion of a rational number ξ satisfying inequality (1).

Theorem 2.1. (Legendre's Theorem.) *Let ξ be a rational number and y and z be positive integers such that gcd$(y, z) = 1$. Suppose*

$$\left|\xi - \frac{y}{z}\right| < \frac{1}{2z^2}, \tag{1}$$

then $\frac{y}{z}$ is a convergent of the continued fractions expansion of ξ.

Proof. See Hardy and Wright (1965). □

Lemma 2.1. *Let $N = pq$ be an RSA modulus with $q < p < 2q$. Then*

$$\frac{\sqrt{2}}{2}\sqrt{N} < q < \sqrt{N} < p < \sqrt{2}\sqrt{N}.$$

Proof. See Nitaj (2008). □

Lemma 2.2. *(Bunder and Tonien, 2017) Let $N = pq$ be an RSA modulus with $q < p < 2q$. Let the public exponent e satisfies an equation $ed - k\phi(N) = 1$ where $d, k \in \mathbb{Z}^+$ and $\phi(N) = (p-1)(q-1)$, then $N + 1 - \frac{3}{\sqrt{2}}\sqrt{N} < \phi(N) < N + 1 - 2\sqrt{N}$.*

Proof. Let$N = pq$ with $q < p < 2q$ and $\phi(N) = (p-1)(q-1) = N - (p+q) + 1$. Then, we can write an interval for $N - \phi(N) = (p+q) - 1$ in terms of N. Now we have $N - \phi(N) = (p+q) - 1 = (p + \frac{N}{p}) - 1$. We define a function f such that $f(p) = p + \frac{N}{p} - 1$. Then, the derivative of f is

$$f'(p) = 1 - \frac{N}{p^2} = 1 - \frac{pq}{p^2} = 1 - \frac{q}{p} > 0 \tag{2}$$

From (2), it shows that the function f is strictly increasing on interval $p \in (\sqrt{N}, \sqrt{2}\sqrt{N})$. Hence, $f(\sqrt{N}) < f(p) < f(\sqrt{2}\sqrt{N})$ leads to

$$\begin{aligned}\sqrt{N} + \frac{N}{\sqrt{N}} - 1 < p + \frac{N}{p} - 1 &< \sqrt{2}\sqrt{N} + \frac{N}{\sqrt{2}\sqrt{N}} - 1\\ 2\sqrt{N} - 1 < p + q - 1 &< \frac{3}{\sqrt{2}}\sqrt{N} - 1\\ 2\sqrt{N} < p + q &< \frac{3}{\sqrt{2}}\sqrt{N}\\ 2\sqrt{N} - 1 < N - \phi(N) &< \frac{3}{\sqrt{2}}\sqrt{N} - 1 \qquad (3)\end{aligned}$$

Since $p + \frac{N}{p} - 1 = p + q - 1 = N - \phi(N)$, then from (3), we have the following relation

$$N + 1 - \frac{3}{\sqrt{2}}\sqrt{N} < \phi(N) < N + 1 - 2\sqrt{N}$$

This terminates the proof. □

Then, we apply the result of Lemma 2.2 and propose the continuous midpoint subdivision analysis upon the interval containing $\phi(N) = (p-1)(q-1)$. Note that, we have previously proposed the same method to construct attacks on the variants of RSA as published in Ruzai et al. (2020).

Based on Lemma 2.2, let $\phi(N) \in (\theta_1, \theta_2)$ where $\phi(N) = (p-1)(q-1)$, $\theta_1 = N + 1 - \frac{3}{\sqrt{2}}\sqrt{N}$ and $\theta_2 = N + 1 - 2\sqrt{N}$. Next, we divide equally the interval (θ_1, θ_2) to obtain a midpoint term denoted as $\mu_{(0,0)}$ as illustrated in Figure 1. This process is denoted with $i = 0$.

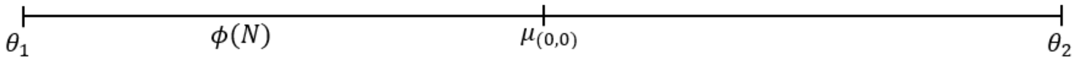


Figure 1

Notice here, no matter where $\phi(N)$ lies on the interval, we always have

$$|\phi(N) - \mu_{(0,0)}| < \frac{\theta_2 - \theta_1}{2}.$$

Continuing, we divide equally between the midpoints of the above intervals; $(\theta_1, \mu_{(0,0)})$ and $(\mu_{(0,0)}, \theta_2)$, which yields another midpoints denoted $\mu_{(1,0)}$ and $\mu_{(1,1)}$ as illustrated in Figure 2.

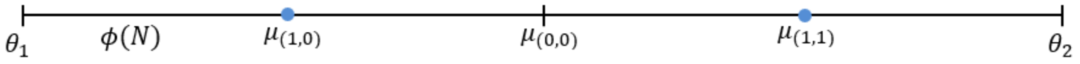


Figure 2

Note that, this process is the first division between the midpoints and we denote with $i = 1$. Then, no matter where $\phi(N)$ lies on the interval, we always have

$$|\phi(N) - \mu_{(1,j)}| < \frac{\theta_2 - \theta_1}{4}, \quad 0 \le j \le 1.$$

Continuously after the first division between midpoints, we equally divide between midpoints as illustrated in Figure 3 and denoted the process with $i = 2$.

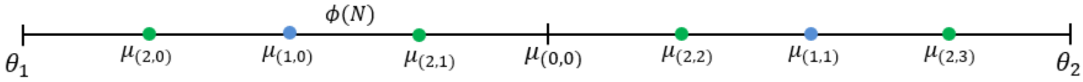


Figure 3

Here, the midpoints obtained from the second division of midpoints are denoted with $\mu_{(2,0)}$, $\mu_{(2,1)}$, $\mu_{(2,2)}$ and $\mu_{(2,3)}$. Then, no matter where $\phi(N)$ lies on the interval, we always have

$$|\phi(N) - \mu_{(2,j)}| < \frac{\theta_2 - \theta_1}{8}, \quad 0 \leq j \leq 3.$$

Similarly, the process continues and the midpoints obtained from the third divison between the previous midpoints are denoted as follows; $\mu_{(3,0)}$, $\mu_{(3,1)}$, $\mu_{(3,2)}$,$\mu_{(3,3)}$, $\mu_{(3,4)}$, $\mu_{(3,5)}$, $\mu_{(3,6)}$ and $\mu_{(3,7)}$. This process is illustrated in Figure 4 and denoted with $i = 3$.

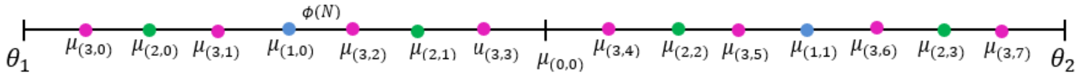


Figure 4

Therefore, regardless of where the interval of $\phi(N)$ may belong, we always have

$$|\phi(N) - \mu_{(3,j)}| < \frac{\theta_2 - \theta_1}{16}, \quad 0 \leq j \leq 7.$$

In summary, we obtain the following general result.

Definition 2.2. *Let $\phi(N) \in (\theta_1, \theta_2)$ where $\phi(N) = (p-1)(q-1), \theta_1 = N + 1 - \frac{3}{\sqrt{2}}\sqrt{N}$ and $\theta_2 = N + 1 - 2\sqrt{N}$. Let i and j be fixed positive integers for the midpoint term $\mu_{(i,j)}$ given by*

$$\mu_{(i,j)} = N + 1 - \sqrt{N}\left(\frac{1+2j}{2^i} + \frac{3(2^{i+1}) - 3 - 6j}{2^{i+2}}\sqrt{2}\right),$$

then

$$\left|\phi(N) - \mu_{(i,j)}\right| < \frac{\theta_2 - \theta_1}{2^{i+1}}.$$

for the specific $\mu_{(i,j)}$.

Remark 2.1. *One can view $i \in \mathbb{Z}$ as the number of subdivision process between the midpoints in the interval of $\phi(N)$ where $\phi(N) \in (N + 1 - \frac{3}{\sqrt{2}}\sqrt{N}, N + 1 - 2\sqrt{N})$ while $j \in \mathbb{Z}$ denotes each midpoint term in i-th subdivision process.*

3 ANALYSIS AND DISCUSSION

In this section, we present our main result. Remark that Theorem 3.1 can be considered as an improved result in terms of the upper cryptanalytic bound of private exponent d as opposed to previous results in Bunder and Tonien (2017) and Tonien (2018).

Theorem 3.1. *Suppose i is a fixed positive integer. Consider an RSA cryptosystem with public key pair (N, e) such that $N = pq$ where $q < p < 2q$. If $e < (p-1)(q-1)$ satisfies an equation $ed - k(p-1)(q-1) = 1$ for some positive integers k and d with*

$$d < \sqrt{\frac{2^i \cdot B}{e \cdot (A + 2^{i+2})}},$$

where $A = (\frac{3}{\sqrt{2}} - 2)\sqrt{N}$ and $B = N^2 - \frac{6}{\sqrt{2}}N^{1.5} + \frac{9}{2}N$, then $\frac{k}{d}$ can be found among the convergents of the public rational number $\frac{e}{\lfloor \mu_{(i,j)} \rfloor}$ given that $\mu_{(i,j)} = N + 1 - \sqrt{N}\left(\frac{1+2j}{2^i} + \frac{3(2^{i+1}) - 3 - 6j}{2^{i+2}}\sqrt{2}\right)$ for some $j \in [0, 2^i - 1]$.

Proof. Let $\phi(N) = (p-1)(q-1)$ be the Euler totient function. Suppose $\phi(N) \in (\theta_1, \theta_2)$ where $\theta_1 = N + 1 - \frac{3}{\sqrt{2}}\sqrt{N}$ and $\theta_2 = N + 1 - \sqrt{2}N$. Let $\mu_{(i,j)} = N + 1 - \sqrt{N}\left(\frac{1+2j}{2^i} + \frac{3(2^{i+1}) - 3 - 6j}{2^{i+2}}\sqrt{2}\right)$ be the general term for midpoint in the interval of (θ_1, θ_2). Then, for every $\mu_{(i,j)}$ we have

$$\left|\phi(N) - \lfloor \mu_{(i,j)} \rfloor\right| < \frac{\theta_2 - \theta_1}{2^{i+1}} < \frac{\theta_2 - \theta_1}{2^{i+1}} + 1. \tag{4}$$

From equation $ed - k\phi(N) = 1$, divide with $d\phi(N)$ to obtain

$$\frac{e}{\phi(N)} - \frac{k}{d} = \frac{1}{d\phi(N)}.$$

Let $\lfloor \mu_{(i,j)} \rfloor$ be the approximation of $\phi(N)$ and observe

$$\begin{aligned}\left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor} - \frac{k}{d}\right| &= \left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor} - \frac{e}{\phi(N)} + \frac{e}{\phi(N)} - \frac{k}{d}\right| \\ &\leq \left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor} - \frac{e}{\phi(N)}\right| + \left|\frac{e}{\phi(N)} - \frac{k}{d}\right| \\ &\leq \frac{e\left|\phi(N) - \lfloor \mu_{(i,j)} \rfloor\right|}{\lfloor \mu_{(i,j)} \rfloor \phi(N)} + \frac{1}{d\phi(N)}. \end{aligned} \tag{5}$$

Next, since $d = \dfrac{1 + k\phi(N)}{e}$ and from (4), then (5) yields

$$\begin{aligned}\left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor} - \frac{k}{d}\right| &< \frac{e(\frac{\theta_2 - \theta_1}{2^{i+1}} + 1)}{\lfloor \mu_{(i,j)} \rfloor \cdot \phi(N)} + \frac{1}{\frac{1 + k\phi(N)}{e} \cdot \phi(N)} \\ &< \frac{e(\theta_2 - \theta_1 + 2^{i+1})}{2^{i+1} \cdot \lfloor \mu_{(i,j)} \rfloor \cdot \phi(N)} + \frac{e}{\phi(N) \cdot [1 + k\phi(N)]}. \end{aligned} \tag{6}$$

Observe that from Lemma 2.2,
$\theta_1 < \phi(N) < \theta_2 \implies \dfrac{1}{\theta_2} < \dfrac{1}{\phi(N)} < \dfrac{1}{\theta_1}$

will lead (6) to

$$\begin{aligned}\left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor}-\frac{k}{d}\right| < \frac{e(\theta_2-\theta_1+2^{i+1})}{2^{i+1}(\theta_1)(\theta_1)}+\frac{e}{(\theta_1)(\theta_1)} &= \frac{e(\theta_2-\theta_1+2^{i+1}+2^{i+1})}{2^{i+1}(\theta_1)^2}\\ &< \frac{e(\theta_2-\theta_1+2^{i+2})}{2^{i+1}(\theta_1-1)^2}\\ &< \frac{e\left[(\frac{3}{\sqrt{2}}-2)\sqrt{N}+2^{i+2}\right]}{2^{i+1}(N-\frac{3}{\sqrt{2}}\sqrt{N})^2}. \end{aligned} \tag{7}$$

For simplicity, let $A=(\frac{3}{\sqrt{2}}-2)\sqrt{N}$ and $B=N^2-\frac{6}{\sqrt{2}}N^{1.5}+\frac{9}{2}N$.
Now, (7) becomes

$$\left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor}-\frac{k}{d}\right| < \frac{e[A+2^{i+2}]}{2^{i+1}\cdot B}. \tag{8}$$

To ensure (8) satisfies the Legendre's Theorem,

$$\frac{e[A+2^{i+2}]}{2^{i+1}\cdot B} < \frac{1}{2d^2}. \tag{9}$$

Then, we solve for d and yields

$$d < \sqrt{\frac{2^i\cdot B}{e\cdot(A+2^{i+2})}}. \tag{10}$$

If (10) holds, then (8) satisfies the condition of Legendre's Theorem;

$$\left|\frac{e}{\lfloor \mu_{(i,j)} \rfloor}-\frac{k}{d}\right| < \frac{1}{2d^2}.$$

Thus, $\frac{k}{d}$ is amongst the convergents of the continued fractions expansion of $\frac{e}{\lfloor \mu_{(i,j)} \rfloor}$. □

As a consequence, by knowing the values of k and d implies that one can solve for prime factors p and q of modulus N.

Corollary 3.1. *Suppose we acquire the unknowns d and k according to Theorem 3.1, then $N=pq$ can be factored in polynomial time.*

Proof. Based on the relation $\phi(N)=(p-1)(q-1)=\dfrac{ed-1}{k}$ in Theorem 3.1, by solving the roots of quadratic polynomial $x^2-(N-\phi(N)+1)x+N=0$, one can recover the primes p and q of modulus N. □

Remark 3.1. *Based on the result obtained in Theorem 3.1, this method is applicable whenever $e>N^{0.5}$.*

Proof. In Theorem 3.1, we assert that one can solve the factorization of N via the continued fractions algorithm if

$$d < \sqrt{\frac{2^i \cdot B}{e \cdot (A + 2^{i+2})}}$$

where $A \approx \sqrt{N}$ and $B \approx N^2 - N^{1.5} + N$.
First, assume that $e \approx N^{\beta}$. From an equation $ed - k\phi(N) = 1$, we have

$$ed = 1 + k\phi(N) > \phi(N) \approx N.$$

Then

$$d > \frac{N}{e} = N^{1-\beta}.$$

Now, the condition of Theorem 3.1 becomes

$$\begin{aligned} d < \frac{\sqrt{2^i}\sqrt{B}}{\sqrt{e}\sqrt{A+2^{i+2}}} < \frac{\sqrt{2^i}\sqrt{B}}{\sqrt{e}\sqrt{A}} &< \frac{\sqrt{2^i}\sqrt{N^2}}{\sqrt{N^{\beta}}\sqrt{N^{0.5}}} \\ &= \frac{\sqrt{2^i}N}{N^{\frac{\beta}{2}} \cdot N^{0.25}} \\ &= \sqrt{2^i} \cdot N^{0.75-0.5\beta}. \end{aligned}$$

Consequently, this method is not working if

$$\begin{aligned} N^{1-\beta} &> \sqrt{2^i} \cdot N^{0.75-0.5\beta} \\ 1 - \beta &> 0.75 - 0.5\beta \\ \beta &< 0.5. \end{aligned}$$

Thus, if $e < N^{0.5}$, then this method is not applicable. □

Next, we demonstrate numerically the proposed attack based on Theorem 3.1. Here, we consider the case when the Euler's totient function lies in the interval when $i = 10, j = 0$.

Example 3.1. *When $i = 10,\ j = 0$.*
On input of an RSA modulus N and public exponent e according to the conditions stated in Theorem 3.1,

$$\begin{aligned} N &= 35209, \\ e &= 409. \end{aligned}$$

We begin the process of factoring N. Let $\mu_{(10,0)} = N + 1 - \left(\frac{1}{1024} + \frac{6141}{4096}\sqrt{2}\right)\sqrt{N}$, then $\frac{k}{d}$ is found in the list of convergents of continued fractions expansion of $\frac{e}{\lceil \mu_{(10,0)} \rceil}$. The list of the convergents are

$$\left[0, \frac{1}{85}, \frac{8}{681}, \frac{9}{766}, \frac{26}{2213}, \frac{61}{5192}, \frac{87}{7405}, \frac{409}{34812}\right].$$

From the above list, we obtain the candidate for $\frac{k}{d} = \frac{8}{681}$ and compute $\phi(N) = \frac{ed-1}{k}$ which result in

$$\phi(N) = 34816.$$

Upon obtaining the value of $\phi(N)$, we continue to find the roots x_1 and x_2 of polynomial $x^2 - (N - \phi(N) + 1)x + N = 0$; which returns the value of primes $p = x_1$ and $q = x_2$ where in this case, $p = 257$ and $q = 137$. This completes the factorization of N.

Observe that from Example 3.1, we verify that the condition $d < \sqrt{\frac{2^{10} \cdot B}{e \cdot (A + 2^{12})}} \approx 858$ where $A = 23$ and $B = 1211802529$ is met as required by Theorem 3.1. Note that, the upper bounds of d in Bunder and Tonien (2017) and Tonien (2018) will fail to retrieve the primes p and q as $d = 681 > \sqrt{\frac{8N^{1.5}}{e}} \approx 359$ (Bunder and Tonien, 2017) and $d = 681 > \sqrt{\frac{8tN^{1.5}}{e}}$ for $t \leq 3$ (Tonien, 2018).

Remark 3.2. *As a note, if $e \sim N$, then the security bound for d is approximately $N^{0.25}$. This reaffirms the classical result by Wiener (1990).*

Remark 3.3. *In comparison with the previous attacks proposed on RSA cryptosystem as in Bunder and Tonien (2017) and Tonien (2018), we increase the security bound of d exponentially; that is from $d < \sqrt{8}N^{0.25}$ (Bunder and Tonien, 2017) and $d < \sqrt{8t}N^{0.25}$ (Tonien, 2018) to $d < \sqrt{2^i}N^{0.25}$. Note that, t and i are both fixed positive integers.*

Remark 3.4. *According to the latest technological advancement as mentioned in Barker (2016), $i = 112$ is an achievable target. Hence, in our case, if we consider $i = 112$, then we are able to increase the security bound of d up to $d < N^{0.305}$ for 1024-bit of modulus N and $d < N^{0.277}$ for 2048-bit of modulus N. This shows some significant improvement as compared to the previously proposed attacks on RSA.*

4 SUMMARY

In summary, we have extend our proposed method that is the continuous midpoint subdivision analysis upon the RSA cryptosystem. Precisely, we prove that we are able to obtain the unknowns d and k from the list of convergents of the continued fractions expansion of certain related rational numbers with certain conditions. At the end of this work, we remark that we raise the security boundary of private exponent d exponentially as opposed to the previous results with similar approach (i.e. via the continued fractions algorithm).

ACKNOWLEDGMENTS

This research was supported by the Ministry of Education of Malaysia with Fundamental Research Grant Scheme under Grant FRGS/1/2019/STG06/UPM/02/8.

REFERENCES

Barker, E. (2016). Recommendation for Key Management Part 1: General. NIST Special Publication 800-57 Part 1 Revision 4, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Boneh, D. and Durfee, G. (2000). Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46(4):1339–1349.

Bunder, M. and Tonien, J. (2017). A new attack on the RSA cryptosystem based on continued fractions. *Malaysian Journal of Mathematical Sciences*, 11(S3):45–57.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

Hardy, G. and Wright, E. (1965). *An Introduction to the Theory of Numbers*. Oxford University Press, London.

Nitaj, A. (2008). Another generalization of Wiener's attack on RSA. In *International Conference on Cryptology in Africa*, pages 174–190. Springer.

Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communication of the ACM*, 21(2):17–28.

Ruzai, W. N. A., Ariffin, M. R. K., Asbullah, M. A., Mahad, Z., and Nawawi, A. (2020). On the improvement attack upon some variants of RSA cryptosystem via the continued fractions method. *IEEE Access*, 8(1):80997–81006.

Tonien, J. (2018). *Continued Fractions and Their Applications*. PhD thesis, University of Wollongong, Wollongong, Australia.

Wiener, M. J. (1990). Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558.

Review of Blockchain-Based Public Key Infrastructure

Chong-Gee Koa[1], **Swee-Huay Heng**[1], **Syh-Yuan Tan**[2], and **Ji-Jian Chin**[3]

[1]*Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia*
[2]*School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom*
[3]*Faculty of Engineering, Multimedia University, Cyberjaya, Malaysia*

E-mail: cyberkoa@gmail.com
[1]*shheng@mmu.edu.my,* [2]*syh-yuan.tan@newastle.ac.uk,* [3]*jjchin@mmu.edu.my*

ABSTRACT

Public Key Infrastructure (PKI) provides a secure mean of authenticating identities over the Internet. A blockchain is a decentralised transaction and data management technology initially developed for Bitcoin cryptocurrency. The interest in blockchain technology has been increasing since the idea was coined in 2008. Blockchain is the most intriguing technology in the finance industry after witnessing the success of Bitcoin. In this paper, we provide an overview on the development of integrating blockchain technology in PKI to improve or extend on PKI functionalities. More precisely, we investigate several recent research work that make use of blockchain technology to overcome the limitation of current implementations of PKI, both for centralised PKI and decentralised PKI.

Keywords: Blockchain, PKI, analysis

1 INTRODUCTION

Public-key infrastructure (PKI) is an integral part of the security of digital communication. The widespread deployment of PKIs has allowed the growth of critical application such as internet banking and e-commerce. Several research have been done on improving the conventional PKI, however, there are still issues that cannot be resolved. The success of blockchain technology has inspired many studies to propose blockchain-based PKIs to build a secure PKI system (Bano, 2017). In this paper, we review the issues of conventional PKIs and the evolution of blockchain-based PKIs. We review several blockchain-based PKIs research chronologically since 2014 and to the best of our knowledge, this is the first analysis paper that reviews the bottleneck of conventional PKIs and the evolution of blockchain-based PKI in recent years. Additionally, we look into the trend and possible future research work on blockchain-based PKIs.

1.1 Blockchain Technology

Blockchain is the core technology used to create the cryptocurrency, Bitcoin, through the maintenance of immutable distributed ledgers proposed by Satoshi Nakamoto (2008). Its potential applications are much wider besides its main usage as an alternative currency. Blockchain technology has been considered as part of the fourth industrial revolution. The use of blockchain as a secured, decentralised and encrypted public ledger has been applied in many areas such as finance, judiciary, commerce and education.

Since the success of Bitcoin and blockchain technology, several improvements have been done on blockchain technology. One of the ideas is to add the scripting capability into blockchain to allow distributed applications in a form of smart contract to run on blockchain. Ethereum (Buterin, 2013), an open source, public blockchain platform was launched in 2015 and rapidly gained the attraction of researchers and developers.

2 ISSUES OF CONVENTIONAL PKI

Public-key cryptography or asymmetric cryptography is a cryptographic system that uses a pair of public and private keys. A PKI manages these keys based on certificates that verify the ownership of a public key by some entity. PKI must support five functionalities, i.e., registration, updating, lookup, verification and revokation (Bano, 2017).

The most common approaches that are used for conventional PKIs fall into two categories, i.e., Certificate Authority (CA) and Webs of Trust (WoT) (Fromknecht et al., 2014). Conventional PKI, whether CA-based or WoT is still open to some issues.

2.1 Certificate Authorities (CAs)

CAs are trusted parties who issue a signed certificate, usually using the standard X.509, to verify an entity's ownership of a public key upon request as in Figure 1. To trust a CA, a device accepts a root certificate for that CA into its storage. A hierarchical certificate chain stems from this root, in which any certificates signed using a trusted certificate are also trusted. For instance, when a user logs into Instagram via a web browser, the web browser will first validate the claimed certificate which holds Instagram's public key by looking into the CA of the given certificate. Often, web browsers are pre-configured to accept certificates from some known CAs.

2.2 PGP Web of Trust (WoT)

PGP WoT approach is totally decentralised, allowing users to designate others as trustworthy by signing their public key certificates (Ryabitsev, 2014). In other words, members of the decentralised network establish trust by verifying that others have a certificate signed by an entity in

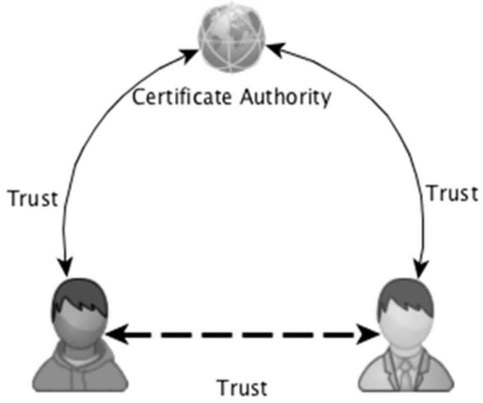


Figure 1: Certificate Authority Managing the Trust between 2 Parties.

whom the verifier has previously established trust. In contrast to CA-based PKI, trust is decentralised in WoT. This means certificate issuance is able to be performed by any party.

3 ISSUES IN CONVENTIONAL PKI

The conventional PKI, whether CA-based or WoT is still open to some issues. Among them are:

- Single point of failure in CA-based PKI (Ellison and Schneier, 2000).
- Certificate Revocation List (CRL) update delay (Lewison and Corella, 2016).
- Lack of identity retention (Fromknecht et al., 2014).
- Lack of incentive on controlling misbehaviour (Matsumoto and Reischuk, 2017).
- Vulnerable to split-world attack (Chuat et al., 2015).

One of the obvious issues is single point of failure (SPOF). It happens in CA-based PKI when the CA as the trusted third party is compromised. This was demonstrated by the DigiNotar case (Prins and Cybercrime, 2011). When the CA is compromised, the attacker can issue fake certificates, has the ability to impersonate certain domains and intercept decrypted traffic through man-in-the-middle (MITM) attacks (Soghoian and Stamm, 2012). Another issue in conventional PKI is lacking of identity retention. In short, it does not guarantee the consistency and nothing prevents different users from generating public keys for the same identity. Besides, in conventional PKI, when a certificate is revoked before their expiration date by CA, it will be kept in CRL. There are a few reasons a certificate might get revoked. In real-world examples, three common reasons a certificate can be revoked are:

1. Private key is lost or compromised.

2. The previous owner of a domain no longer owns the domain.

3. A certificate was found to be imitated fraudulently.

CRL is essentially a large list of blacklisted certificates maintained by CAs. CRL maintenance could be a difficult task. It requires continuous changes and updates hence is susceptible to errors. Therefore, delay on update of CRL is a common issue of conventional PKI. Online Certificate Status Protocol (OCSP) was born as the alternative for the CRL to mitigate the threat of the CRL update delay. An OCSP server is acting as an online responder to query whether a certificate is already revoked or not. It will respond with good, revoke or unknown to query from OCSP clients. However, OCSP is still open to replay attacks where a signed response is captured by man-in-middle and replayed to the client at a future date even though the subject certificate may have been revoked.

Transport Layer Security (TLS) PKI is the real-world example of the conventional PKI that used in securing the encrypted client-server communication in World Wide Web: HTTPS (Rescorla, 2000). CAs play an important role to ensure the users are browsing the correct website by issuing the certificate to the correct party. It is observed that there are insufficient incentives for CAs to invest more to improve in security (Fromknecht et al., 2014) because CAs gain few rewards for reputation of security and face little consequences for misbehaving (Asghari et al., 2013). Therefore, lacking of incentives to control the misbehaviour of CAs is one of the issues in conventional PKI. Certificate Transparency (CT) was proposed by Google in 2013 to improve the security of conventional TLS PKI with the aim to detect fraudulent TLS certificates which are valid technically. CT provides append-only, publicly auditable logs for all issued certificates, and reduces the certificates lifetime (Wang et al., 2018). The authors in (Mazires and Shasha, 2002) showed that if an attacker can get fake certificates to launch a MITM attack, then the attacker may also be able to control the log and provide the targeted victims with a view that includes a specific certificate only. This attack is later named as split-world attack by Chuat et al. (2015).

Figure 2 summarises the issues of conventional PKI.

4 THE EVOLUTION OF BLOCKCHAIN-BASED PKI

A full-fledged blockchain-based PKI was first introduced by Fromknecht et al. (2014). The authors leveraged the advantages of blockchain technology to build a secure decentralised PKI (DPKI) with identity retention. Compared to conventional DPKI like WoT, blockchain-based PKI improves over the WoT with consistency offering of identity retention. Instead of trusting a third party with a small set of members as in the WoT, the implementation of blockchain-based PKI by Fromknecht et al. only required that users trust that the majority of other users are not malicious. They proposed a blockchain-based PKI, Certcoin which can efficiently support each PKI functionality with blockchain using a Merkle accumulator as CRL.

Leiding et al. (2016) introduced Authcoin a new alternative protocol for authentication with a flexible challenge and response scheme. The authors outlined that the process of authentica-

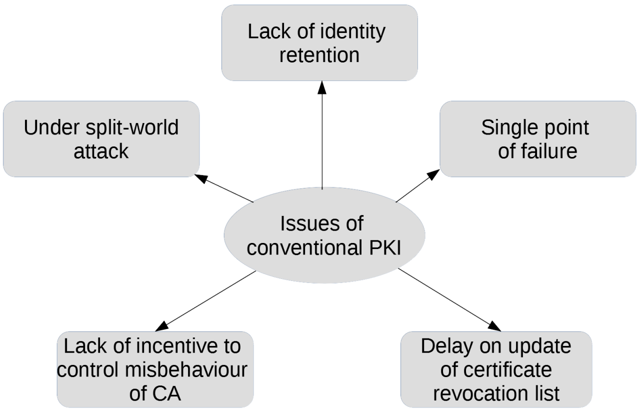

Lack of identity retention
Under split-world attack
Single point of failure
Issues of conventional PKI
Lack of incentive to control misbehaviour of CA
Delay on update of certificate revocation list

Figure 2: Issues of Conventional PKI.

tion starts with key pair generation and binding. After binding, the authors further explained the flow of ”Formal Key Validation” which leads to validation and authentication in AuthCoin. Finally, they discussed the revocation and expiration of key and signature. Compared to Certcoin (Fromknecht et al., 2014), Leiding et al. claimed that Authcoin focused on validation and authentication process to fight malicious users. A custom blockchain was used by them to demonstrate the proposed protocol, and the same protocol can be implemented in any blockchain such as Bitcoin or Ethereum.

The research on blockchain-based systems is getting popular and more improvements have been proposed to improve existing blockchain-based PKI. For instance, Lewison and Corella (2016) proposed special certificates to be used in their proposed blockchain-based PKI. The two new certificate formats, i.e., plain format and rich format, which are designed specifically for blockchain-based PKI, comprise a public key, meta data, and asserted data but no signature. The proposed blockchain-based PKI claimed to solve a longstanding problem of conventional PKIs by not requiring the issuance of CRLs or responding to Online Certificate Status Protocol (OCSP) queries. Besides, since the certificate is not signed, this means it is smaller in size. Therefore, the time taken to transmit a certificate backed by a CA certificate chain is reduced.

Al-Bassam (2017) realised a smart contract-based PKI and identity system (SCPKI) on Ethereum with the target to detect the issuance of fraudulent certificates. He implemented a working prototype of PKI using smart contract that can simulate the functionalities of conventional PKI. He proposed two versions of smart contract, a full version and a light version. Instead of storing the attribute data within smart contract in full version, light version stores attribute data off the Ethereum blockchain, into InterPlanetary File System (IPFS). By utilising IPFS, the gas costs associated with Ethereum storage is lower in light version compared to full version.

Yakubov et al. (2018) proposed a blockchain-based PKI framework that manages the standard X.509 certificates. Instead of creating a new certificate format, they proposed a hybrid certificate by adding blockchain meta data in the X.509v3 extension fields. There are five additional information fields added into the extensions field of X.509v3 certificate, i.e., subject key identifier, blockchain name, CA key identifier, Issuer CA identifier and hashing algorithm. The proposed hybrid certificate and framework provides a transition path from large deployments of conventional CA-based PKI to blockchain-based PKI in practical.

Axon and Goldsmith (2017) continued working to improve blockchain-based PKI by adding the privacy-awareness capability based on the blockchain PKI proposed by Fromknecht et al. (2014). Privacy aware blockchain PKI (PB-PKI) which was proposed by these authors defined three levels of privacy: total anonymity, neighbour group anonymity and user-controlled disclosure. There is always a trade-off between security of PKI and the privacy level of PKI. The better the privacy, the weaker the security. For instance, security is weaker in the case of total anonymity, and neighbour group anonymity provides better security properties. Besides that, Axon and Goldsmith (2017) also presented enhancements on key updates and recovery, revocation and tracing on Certcoin (Fromknecht et al., 2014) in the PB-PKI model.

In another research, Matsumoto and Reischuk (2017) designed a blockhain-based PKI call Instant Karma PKI (IKP). IKP is an automated platform that can incentivise correct behaviour of CA and report the unauthorised certificates automatically. Matsumoto and Reischuk (2017) presented the design of IKP, together with a framework for reacting to CA misbehaviour. They also demonstrated an economic analysis on IKP to check the incentivisation and disincentivisation on CA.

Patsonakis et al. (2017) reviewed CertCoin and proposed improvements on it. They noticed that the blockchain-based PKI that presented by Fromknecht et al. did not exploit sufficiently the potential for two reasons:

1. The state of the system is still of logarithmic complexity because of using a Merkle tree-based accumulator.
2. The construction of recomputing values to handle revocation still has a linear computational complexity.

They proposed a public-state, additive, universal accumulator which is based on the strong RSA assumption in the random oracle model.

Wang et al. (2018) proposed an idea to improve certificate transparency and limited-grained revocation transparency. They proposed a blockchain-based scheme to construct append-only logs for certificate transparency. In this scheme, web servers publish their CA-signed certificates by their subjects in a global certificate blockchain. This blockchain acts as an append-only public accessible log to monitor CAs' operations. Web servers form a community to monitor the certificate published by CAs. Supported with analysis and experimental result, they concluded that, the scheme introduces reasonable overheads in terms of storage, certificate validation delay, communication and incentive cost.

CertLedger was presented by Kubilay et al. (2018). They reviewed several previous attempts to distribute absolute trust on CAs. Kubilay et al. claimed that all of the previous attempts do not resist the split-world attack. CertLedger, on the other hand, made split-world attacks impossible because all Transport Layer Security (TLS) clients can verify the final state of the log thanks to the immutable aspect of the blockchain. They proposed a more transparent revocation process, a unique, efficient and trustworthy certificate validation process in this architecture.

In another research, Chen et al. (2018) designed a public and efficient certificate audit scheme for TLS connections based on blockchain called CertChain. Chen et al. (2018) proposed a new distributed dependability-rank based consensus protocol to be used in CertChain to avoid centralisation in practice issue that happened in common blockchain consensus. To overcome mandatory traversal issue of blockchain, CertChain introduced a new data structure called CertOper that kept in block for operations forward traceability and efficient query. In the proposed solution, CertChain utilised Dual counting bloom filter (DCBF) to achieve real-time certificate validation to overcome block size limitation. CertChain system model comprises four kind of entities: client, domain, CAs and bookkeepers, where bookkeeper is a new entity that proposed to record the certificate operations in the model.

Research of blockchain-based PKI is not only limited to the improvement over conventional PKI. Pinto et al. (2018) proposed a model to use blockchain-based PKI to improve the trust, confidentiality and privacy of Internet of Things (IoT). They pointed out that there is a need to implement a PKI in the IoT network to manage securely the identity of each node in the network. A distributed infrastructure which is able to register the devices admitted in a network into a verifiable and safe data structure has been presented. The infrastructure consists of two components, i.e., identity management and blockchain-based PKI. Experiments had been carried out with several use cases and Pinto et al. (2018) observed that the blockchain-based PKI shows the potential to overcome the WoT PKI. The reason is blockchain-based PKI does not require an interconnected structure of authenticated entities similar to WoT model.

In recent research, Yao et al. (2019) found out that there are still open issues in CertChain that remained unresolved in a large scale implementation. They proposed PBCert (Yao et al., 2019), a privacy-preserving blockchain-based certificate status validation towards massive storage management. In PBCert, all the revoked certificates are stored in OCSP servers while the control information regarding revoked certificates are kept in blockchain. Besides, in order to preserve client privacy, an efficient obscure response to revocation query had been designed in this research. Yao et al. (2019) developed a prototype and they had done several performance analysis on PBCert prototype and CertChain. A comparison between PBCert and CertChain was presented and it is observed that PBCert had reduced the storage size of block over CertChain and at the same time preserved browsing privacy of client.

Blockchain-based PKI still gains the interest from the researchers and Li et al. (2019) had come out with an Internet Web Trust System based on smart contract (Li et al., 2019). They had made an enhancement on SCPKI (Al-Bassam, 2017). In the proposed system, they classified nodes into three types of nodes: CA nodes, end-user nodes and ordinary nodes. The CA nodes are responsible for certificate authentication while end-user nodes are the consumers in the system who will apply for certificate and make query on the certificate. The ordinary nodes are only allowed to view certificates and are involved in the maintenance of blockchain. All the five parts are realised with smart contracts. They also implemented new functions to build an automatic response, reward and punishment mechanism.

5 OUR OBSERVATIONS AND POTENTIAL RESEARCH DIRECTIONS

We observe that due to features of decentralisation, traceability, immutability, and currency property in blockchain, the open issues of conventional PKI such as single point of failure and delay of CRL update and notification, split-world attack are eliminated. Besides, currency properties in blockchain have provided a platform that can incentivise the correct behaviour of CA and also disincentivise the CA for misbehaviour which subsequently improves on the operation of certificate management.

In this paper, we investigate the open issues of the conventional PKI and review several research on blockchain-based PKI and see how they overcome the issues. In Table 1, we chronologically list all the blockchain-based PKI reviewed with the type of blockchain which their proposal was implemented. We observe that most of the latest research on blockchain-based PKI chose Ethereum (Buterin, 2013) as an implementation platform instead of reinventing a new blockchain. We believe that more and more research on blockchain-based PKI will be done on Ethereum.

Blockchain-based PKI	Year	Custom Blockchain	Bitcoin based	Ethereum based
Certcoin (Fromknecht et al., 2014)	2014		x	
Blockstack (Ali et al., 2016)	2016		x	
Authcoin (Leiding et al., 2016)	2016	x		
Pomcor (Lewison and Corella, 2016)	2016	x		
SCPKI (Al-Bassam, 2017)	2017			x
Hybrid X.509v3 (Yakubov et al., 2018)	2017			x
PB-PKI (Axon and Goldsmith, 2017)	2017		x	
IKP (Matsumoto and Reischuk, 2017)	2017			x
Patsonakis PKI (Patsonakis et al., 2017)	2018			x
Wangs PKI (Wang et al., 2018)	2018	x		
CertLedger (Kubilay et al., 2018)	2018			x
CertChain (Chen et al., 2018)	2018			x
Pintos PKI (Pinto et al., 2018)	2018			x
PBCert (Yao et al., 2019)	2019			x
Internet Web Trust System (Li et al., 2019)	2019			x

Table 1: Type of Blockchain for Blockchain-Based PKI

We summarise in Table 2 a list of features for all the blockchain-based PKI reviewed. It is observed that most of the proposed blockchain-based PKI are not suitable for large scale deployment mainly due to problems of blockchain itself. The first problem is block size limitation and the second problem is data query performance which comprises the main functions involved in the revocation and recovery process. In order to overcome these problems, an external solution such as IPFS that is used in SCPKI or an integration of conventional PKI component such as

OCSP can be one of the possible solutions.

Blockchain-based PKI	Privacy preserved	Incentivise mechanism	Conventional PKI compatilibity	Large scale deployment	Smart contract
Certcoin (Fromknecht et al., 2014)	x				
Authcoin (Leiding et al., 2016)	x				
Pomcor (Lewison and Corella, 2016)	x				
SCPKI (Al-Bassam, 2017)	x				x
Hybrid X.509v3 (Yakubov et al., 2018)			x		x
PB-PKI (Axon and Goldsmith, 2017)	x				
IKP (Matsumoto and Reischuk, 2017)		x			x
Patsonakis PKI (Patsonakis et al., 2017)	x				x
Wangs PKI (Wang et al., 2018)	x			x	
CertLedger (Kubilay et al., 2018)	x				x
CertChain (Chen et al., 2018)	x				x
Pintos PKI (Pinto et al., 2018)	x			x	x
PBCert (Yao et al., 2019)	x			x	x
Internet Web Trust System (Li et al., 2019)		x			x

Table 2: Feature Comparison of Blockchain-Based PKI

We also observe and gather some potential research directions of blockchain-based PKI based on our review which focus on the following aspects, as shown in Figure 3.

From the observation of potential research directions, we believe there is still room to improve the existing blockchain-based PKI on privacy preservation capabilities. In real-world applications, privacy preservation is a main feature to control permission on accessing of specific data. The research on privacy aware blockchain-based PKI can be further continued to improve on the PB-PKI (Axon and Goldsmith, 2017) with a permission level and permitted period. Besides, it is believed that further enhancements on new PKI architectures focuses on the implementation of CRL with different types of cryptography accumulators can reduce the storage usage in the blockchain, hence improve the CRL look up performance. Besides, as the IoT is rising in popularity, the demand for large scale deployment of blockchain-based PKI can be an interesting research topic. For consensus improvement, we observe that not many research focus in this direction. Among all the blockchain-based PKIs that we reviewed, only CertChain (Chen

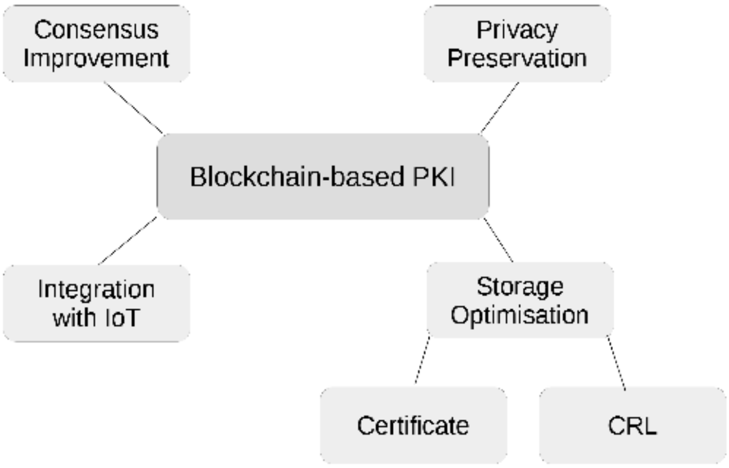


Figure 3: Potential Research Directions on Blockchain-Based PKI

et al., 2018) proposed an improvement with distributed dependability-rank based consensus. Therefore, we believe that further work can be done in this direction. Finally, the limitations of blockchain on the block size and data traversal opens a new direction of research by combining blockchain with other technology to implement a new generation of blockchain-based PKI.

6 CONCLUSION

In summary, we reviewed recent research on blockchain-based PKIs which utilise blockchain technology to overcome the open issues of conventional PKI. We made comparisons among them in several aspects to analyse the pros and cons and features of each proposal. Besides, we also observed the research directions in the recent research of blockchain-based PKI which is useful for the researchers to propose future enhancements.

ACKNOWLEDGEMENT

The authors would like to acknowledge the Multimedia University's GRA scheme and the Malaysia government's Fundamental Research Grant Scheme (FRGS/1/2018/ICT04/MMU/01/01) and Prototype Research Grant Scheme (PRGS/2/2019/ICT04/MMU/01/1) for supporting this work.

REFERENCES

Al-Bassam, . (2017). Scpki: A smart contract-based pki and identity system. In *Proc. ACM Workshop Blockchain Cryptocurrencies Contracts (BCC)*, pages 35–40.

Ali, J., Nelson, R., Shea, M., and Freedman (June 22-24, 2016). Blockstack: a global naming and storage system secured by blockchains. In *Proceedings of the 2016 USENIX Conference on Usenix Annual Technical Conference*, Denver, CO, USA.

Asghari, H., Eeten, M. V., Arnbak, A., and Eijk, N. V. (November 2013). Security econnomics in the https value chain.

Axon, L. and Goldsmith, M. (2017. Jul 24-26, 2017. 2017). Pb-pki: A privacy-aware blockchain-based pki. In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, volume 4, pages 311–318, Madrid, Spain. SECRYPT.

Bano, S. (2017). Consensus in the age of blockchains. Technical report.

Buterin, V. (2013). Ethereum: A next-generation smart contract and decentralized application platform.

Chen, J., Yao, S., Yuan, Q., He, K., Ji, S., and Du, R. (Apr. 2018). Certchain: Public and efficient certificate audit based on blockchain for tls connections. In *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pages 2060–2068.

Chuat, L., Szalachowski, P., Perrig, A., Laurie, B., and Messeri, E. (2015). Efficient gossip protocols for verifying the consistency of certificate logs. pages 415–423.

Ellison, C. and Schneier, B. (2000). Ten risks of pki: What you're not being told about public key infrastructure. *Computer Security Journal*, 16(1):1–7.

Fromknecht, C., Velicanu, D., and Yakoubuv, S. (2014. November 2014). A decentralized public key infrastructure with identity retention. *Cryptology ePrint Archive*, 803.

Kubilay, M., Kiraz, M., and Mantar, H. (Jun. 2018). Certledger: A new pki model with certificate transparency based on blockchain.

Leiding, B., Cap, C., Mundt, T., and Rashidibajgan, S. (September 2016). Authcoin: Validation and authentication in decentralized networks.

Lewison, K. and Corella (2016). Backing rich credentials with a blockchain pki. Technical report, Pomcor.

Li, Wang, N., Du, X., and Liu, A. (2019). Internet web trust system based on smart contract. *Communications in Computer and Information Science*, 1058.

Matsumoto, . and Reischuk, R. (May 2017). Ikp: Turning a pki around with decentralized automated incentives. In *Proc. IEEE Symp. Secur. Privacy (SP)*, pages 410–426.

Mazires, D. and Shasha, D. (2002). Building secure file systems out of byzantine storage. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 108–117. ACM.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Patsonakis, C., Samari, K., Roussopoulos, M., and Kiayias, A. (2017). Towards a smart contract-based, decentralized, public-key infrastructure.

Pinto, G., Dias, J., and Ferreira, H. (Dec. 2018). Blockchain-based pki for crowdsourced iot sensor information. In *Proc. 10th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, pages 248–257.

Prins, J. and Cybercrime, B. (2011). Diginotar certificate authority breach operation black tulip.

Rescorla, E. (May 2000). Https over tls, rfc2818.

Ryabitsev, K. (2014). Pgp web of trust: Core concepts behind trusted communication.

Soghoian, C. and Stamm, S. (2012). Certified lies: Detecting and defeating government interception attacks agains ssl.

Wang, Z., Lin, J., Cai, Q., Wang, Q., Jing, J., and Zha, D. (2018). Blockchain-based certificate transparency and revocation transparency. in financial cryptography and data security.

Yakubov, A., Shbair, W., Wallbom, A., Sanda, D., and State (2018). R: A blockchain-based pki management framework. In *Proceedings of 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan.

Yao, S., Chen, J., He, K., Du, R., Zhu, T., and Chen, X. (2019). Privacy-preserving blockchain-based certificate status validation toward mass storage management.

Efficient Distributed Key Agreement for Edge Devices in the Internet of Things with Information-Theoretic Security

Abid Rauf[*1], **Zhaohong Wang**[2], **Hasan Sajid**[3], and **Muhammad Ali Tahir**[1]

[1]*Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan*
[2]*Department of Electrical and Computer Engineering, California State University, Chico*
[3]*Department of Robotics and Intelligent Machine Engineering, National University of Sciences and Technology, Islamabad Pakistan*

E-mail: abid.rauf@seecs.edu.pk
[*]*Corresponding author*

ABSTRACT

The Internet of Things (IoT) has a broad application in various domains, from small scale smart homes to large scale smart grids and smart cities. As different devices equipped with the capability of connection to communication networks, the weakness of IoT edge devices becomes a vulnerability of the entire system. Adversaries try to infiltrate into the communication between IoT edge devices and then launch attacks on the bigger system. Besides, the IoT edge devices generally lack computational resources compared to traditional computers, making IoT edge devices inefficient in carrying out computational-intensive cryptographic tasks. Therefore, how to update keys efficiently becomes a challenge throughout the life span of those IoT edge devices. In this paper, we propose a secret key establishment scheme suitable for securing IoT comprised of heterogeneous edge devices. By forming closely located IoT edge devices into enclaves of devices, our proposed framework achieves distributed secret generation in an information-theoretic secure sense. Simulations validated our proposed design in terms of its effectiveness and run-time complexity.

Keywords: key agreement, heterogeneous Internet of Things, information-theoretic security

1 INTRODUCTION

Internet of Things (IoT) finds its footage in many domains. As hardware cost for embedded systems becomes more and more affordable, many manufacturers updated their products with the capability of connecting to the Internet. The newly upgraded products become the “edge” of the Internet of Things, identified as “edge devices”. An important type of IoT is large scale IoT that comprises heterogeneous edge devices, usually found in critical infrastructures. Examples of such IoT include smart communities, smart cities, and smart grids. However, end-user devices that are at the edge of the IoT usually come from different vendors. Specifically, for critical infrastructures such as smart grids and smart cities, edge devices from different vendors vary significantly. Take the smart inverters from smart grids, for example. Researchers in Germany recently discovered an IoT formed by heterogeneous commercial energy storage systems with security flaws (Baumgart et al., 2019). The smart city is another example where the security of its IoT has a significant impact on the physical infrastructure. Adversaries aim to cause physical damage through low-cost cyberspace attacks (AlDairi et al., 2017). Therefore, it is a concern that large IoT comprised of heterogeneous edge devices expose vulnerabilities for adversaries. A necessary countermeasure is to secure the communications among entities in IoT with heterogeneous edge devices.

Classical public-key cryptosystems rely on the assumption that adversaries do not have enough computational resources to crack the system in a short time. With the advent of more and more powerful computing machines, old systems’ computational security is at risk. The threat, however, is real. Cryptographic primitives based on the hardness of mathematical problems, including the public key schemes, are well-known to succumb to powerful computing machines (Del Pino et al., 2017). In the advent of quantum computers, continuing to use classical public-key primitives developed decades ago before researchers had developed quantum algorithms that already opens doors for attackers to crack the implementation Cheng et al. (2017), Mosca (2018). With quantum computers, attackers can decrypt classical encryption exponentially faster than classical computers (Cheng et al., 2017, Mosca, 2018).

With the awareness of powerful adversaries, our work aims to address the secure communication challenges observed in the rising situation of heterogeneous IoT. While we do not discard classical public-key schemes, we would like to propose an alternative approach that does not rely on the assumption of computational security. Therefore, adversaries with quantum computing ability will not gain an advantage in breaking our scheme. In this paper, we propose a distributed scheme for achieving secret key agreement through the formation of enclaves of IoT devices against external adversaries. The scheme is scalable to IoTs with the ever-growing edge devices - in each enclave, the data server forms an intranet with edge devices. The goal of the proposed scheme is to efficiently manage the secret keys throughout the lifespan of the IoT devices to secure communication with the data servers against external adversaries. The significant contribution and innovations are as follows. Our scheme separates the communication with the data server from other multi-modal accesses. Then legitimate nodes generate and update secret keys efficiently compared with existing schemes without a centralized server. As such, our proposed scheme helps exclude external adversaries from communications inside the IoT involving edge devices. The other major contribution of our work is the Information security proof of our scheme.

The remaining of the paper is organized as follows. Related work regarding IoT communication security is reviewed in Section 2. A formal description of the adversary model is discussed in Section 3. Our detailed proposed scheme is described in Section 4. Security analysis of the proposed scheme is carried out in Section 5. Experiments and simulations are described in

Section 6 and we conclude in Section 7.

2 RELATED WORK

In this section, we briefly overview previous work that relate to the building blocks in our proposed key generation scheme.

A related area is the secret key establishment schemes for wireless sensor networks (WSN). The key establishment in WSNs mainly relies on key pre-distribution offline on a large scale (Abdallah et al., 2015, Cheng et al., 2020, Harn and Hsu, 2015, Kumari et al., 2017, Mirvaziri and Hosseini, 2020, Zhang et al., 2020). The traditional Public Key Infrastructure (PKI) is a centralized approach relying on a trusted third-party. It also requires high computation cost for signature generation and verification as it requires the expensive operation of exponentiation. Choi et al. proposed a scheme for random key generation and to manage the public key blockchain Choi et al. (2020). Hsiao et al. proposed the date-constrained hierarchical key management scheme for mobile agents using PKI (Hsiao et al., 2019). Nicanfar et al. proposed a mutual authentication scheme among smart meters in different areas in a hierarchical network through identity-based cryptography (Nicanfar et al., 2014). However, all these schemes demand high computation and communication cost because of the use of PKI.

An alternative is the information-theoretic secure approach. Key generation using Information-theoretic approach is to investigate how to generate a secret key shared between the target terminals from an information-theoretic perspective. Key generation can be achieved under source models (Lai and Ho, 2015, Nitinawarat et al., 2010, Tavangaran et al., 2018, Xu et al., 2016b,c, Ye and Reznik, 2007, Zhang et al., 2014, 2017) and channel models (Csiszár and Narayan, 2008, Lai et al., 2012, 2011, Wang et al., 2013, Xu et al., 2016a, Zhou et al., 2014). In source models, terminals can generate secret keys using correlated sources observed from the outputs of discrete memoryless source (DMS), over a public noiseless channel that an eavesdropper has complete access to (i.e., public discussion Maurer (1993)). In channel models, transmitters and receivers can generate secret keys through noisy channels, such as wiretap channel with and without feedback (Ahlswede and Cai, 2006, Ardestanizadeh et al., 2009, Bassi et al., 2018, Dai and Luo, 2018), and multiple access channel with feedback (Salimi et al., 2013).

Recently, the joint source-channel models for secret key generation have drawn considerable attention, and some interesting models were investigated in (Khisti et al., 2012, Prabhakaran et al., 2012, Salimi and Skoglund, 2012, Tu et al., 2016), where transmitters and receivers wish to share secret keys using correlated sources and noisy channels. More precisely, (Khisti et al., 2012) is of great significance and considered a one receiver, eavesdropper model, where two legitimate terminals observe correlated source sequences from DMS and are connected through a DMC, with an eavesdropper, who has access to the channel completely. The secret key generated by the two legitimate terminals needs to keep secret from the eavesdropper. The full characterizations of secret key capacity were provided when a two-way public noiseless channel is available and not available.

As a further extension of the model in Khisti et al. (2012), reference (Tu et al., 2016) considered concatenating an additional legitimate terminal with the transmitter through a noiseless channel that the eavesdropper has access to. Under this model, the problem of simultaneously generating two keys between the legitimate receiver with the transmitter and the additional terminal, respectively, was considered. The two keys all need to be concealed from the eavesdropper, while the key generated by the additional terminal should be additionally protected from the transmitter, and the key generated by the transmitter needs to additionally keep secret from the additional terminal. All terminals except transmitter can observe correlated source sequences from DMS. The key capacity regions for the two expected keys are established by designing joint source-channel coding schemes to achieve these regions.

Existing information theoretical results characterize the maximum achievable secrecy rate under various channel models (Csiszr and Narayan, 2005, Ekrem and Ulukus, 2009, Maurer, 1993, Wyner, 1975). Wyner pointed out that parties can establish information-theoretic secrecy by using the noisy broadcast of wireless transmissions (Wyner, 1975). The most common setting considers pairwise secret key generation over a single channel with a single sender and one or more receivers. Some results are available for a network setting, most notably secure network coding for an error-free wired network (Cai and Yeung, 2011). Maurer showed the usefulness of feedback from Bob to Alice, even though the feedback channel is available to Eve and the public (Maurer, 1993). Mauer showed that feedback enables Alice and Bob to generate a key in an information-theoretically secure sense against Eve, even if the channel from Alice to Bob is inferior to the channel from Alice to Eve. This line of work has led to a rich set of literature on pairwise unconditional secret-key agreement with public discussion (Kanukurthi and Reyzin, 2009). Csiszar and Narayan studied key agreement among a group of terminals connecting to a noisy broadcast channel (Csiszr and Narayan, 2005). They established some achievable secrecy rates, under the assumption that Eve has no access to the broadcast messages.

Siavoshani et al. designed a scheme for multi-terminal over erasure networks (Siavoshani et al., 2011). The work assumed that Eve also has access to the noisy broadcast channel, but Eve is not present at multiple places in the arena at the same time. To the best of our knowledge, ours is the first work to apply multi-terminal secret key agreement over erasure networks commonly observed in the communication between edge devices in the heterogeneous IoT, such as the smart inverters with the utility grid (Pacific Gas & Electric Company, 2018), where Eve also has access to the noisy broadcast channel and can be present at more than one place but not all places within the arena.

3 PROBLEM STATEMENT, CHALLENGES AND ADVERSARIAL MODEL

In this section, we formulate the adversaries to the communication in the scenario of Internet of Things with heterogeneous edge devices (IoT-HED). The term ”IoT-HED” denotes the edge device in the network in our discussion. The necessary communication between the IoT data server and IoT-HED are the regular control profiles from the IoT data server to the IoT-HEDs daily, and real-time control or inquiries from the IoT data server to IoT-HEDs as needed. We restrict the IoT-HEDs on end-user premises to be enclaves. Naturally, IoT-HEDs in a neighborhood could form an enclave that is entirely independent of other enclaves, as depicted in

Figure 1. The enterprise maintains a communication terminal in each enclave. In an enclave, the IoT data server's terminal and communication terminals of IoT-HEDs form an intranet. As such, the enclaves of IoT-HEDs are at the edge of the enterprise IoT network. We consider the adversaries as external to communication scenarios being passive (Goldreich, 2006). The situation when adversaries already intrude into terminals of the IoT-HEDs trying to impersonate legitimate nodes is out of the scope of this work (Lee, 2013). They may eavesdrop the messages transferred in the communication lines and may try the replay attacks, but they do not actively tamper encrypted messages. We assume that using an appropriate form of hash can detect tampering of messages. The purpose of the adversaries is to decrypt the messages, perhaps further infer the operation of the enterprise to find opportunities for launching sophisticated attacks. We also assume that the adversaries cannot hear every single message in the communication among legitimate terminals, although the adversaries may be present in multiple locations in the network formed by the legitimate terminals. The observation of uptime in the pilot study on an IoT-HED with smart inverters justifies our assumption (Pacific Gas & Electric Company, 2018). In the communication between the IoT data server and the IoT-HED, while legitimate terminals experience inconsistent uptime to receive messages from the IoT data server, the adversary also is not supposed to be "online" at the same time in every single location of the terminals. We do not assume the adversaries' computational ability.

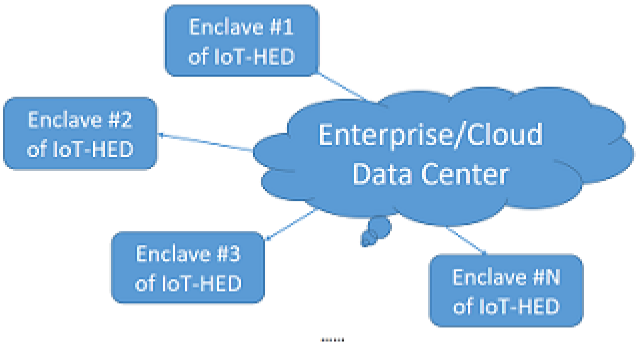


Figure 1: Communication between the Data Center and IoT-HEDs on Enclaves

In the context of IoT-HEDs against the modeled adversaries, the task of securing communication faces the following challenges. **Challenge C1**: update the secret keys among legitimate terminals efficiently throughout the long lifespan of the terminals. Due to significant initial capital investment, the IoT-HEDs are supposed to function for many years, especially in critical infrastructures. Therefore, there is an urgent need to replace secret keys efficiently for a large scale of terminals in their lifetime. **Challenge C2**: manage secret keys in a distributed way among distrusting terminals. The IoT-HED devices are independent of each other, each belong to its owner. Therefore, centralized key registration and management commonly found in the traditional key management mechanisms do not suit in the context of IoT-HED. A single point of failure in the centralized key server may reveal all stored keys. **Challenge C3**: the importance of communication in IoT-HED calls for a higher level of security. The security of public-key cryptography relies on the difficulty of solving the underlying mathematics problem. Cracking the keys may not be so challenging to adversaries with abundant computing resources, not to mention the quantum computing adversaries. Therefore, the traditional PKI with trusted third parties is not suitable in the IoT-HED.

In contrast, information-theoretic security does not rely on computational security based on the difficulty of solving certain mathematical problems. To address the problems and challenges stated in the above, we proposed our information-theoretic distributed key establishment scheme described in Section 4. The rationale of our proposed scheme is to incorporate the physical context of the IoT-HED scenario into its secure key establishment design. Traditionally, the perfect secrecy with a one-time pad is impractical because of the key management issue. However, for the specific scenario of IoT-HEDs in a neighborhood, the perfect secrecy with one-time secret keys becomes feasible because of its small scale. The current practice of the multi-modal access of smart inverters has its vulnerability in cyberspace through the internet. Different from the traditional approach of allowing multi-modal access through one wireless connection, our approach separates the IoT data server access to the IoT-HED through an independent wireless communication network. Our scheme forces the adversary to move out of his/her comfort zone, making the low-cost cyber-only attack meaningless. Eavesdropping the end-user (consumer) premise's internet is useless since the communication between the IoT-HEDs and the IoT data server becomes independent from the consumer premise's internet connection. The adversary must be physically present in the consumer premises, and he/she now faces the information-theoretic security scheme to get the secret key.

4 DISTRIBUTED KEY ESTABLISHMENT THROUGH ENCLAVES OF IOT-HED

The scenario under consideration is the communications between the IoT data server and the IoT-HEDs on end-user (consumer) sites. The IoT-HEDs and at least one terminal from the IoT data server form an enclave, as shown in Figure 1. We denote the participating terminals in an enclave as T_i, for $i = 0, ..., k$.

The overall idea is to allow the *n* participating smart devices T_i in an enclave to continue to send bits simultaneously until the end terminals have agreed on a bit sequence, the secret key. Therefore, participating parties store the secret keys generated in a distributed way, and each enclave is independent of other enclaves. The scheme makes the adversary not able to construct the secret with high probability and is of polynomial complexity for legitimate terminals. Real-time generation of new keys also contributes to "backward" secrecy. The reason is that terminals do not reuse old keys, but new keys as soon as possible. Therefore, obtaining the current key does not help to decrypt past messages.

Our proposed distributed key establishment mechanism addresses the challenges **C1** to **C3** based on an information-theoretic secure scheme against adversaries with potentially unlimited computing power (Siavoshani et al., 2011). There is no single point of failure, as observed in the centralized key pre-registration schemes.

4.1 Notations, Definitions, and Building Blocks

To facilitate the discussion of our proposed scheme, we introduce several notations, definitions, and building blocks first. Our building block to achieve distributed key generation is rooted in the distributed secret generation proposed in Siavoshani et al. (2011). Every terminal in the network has a unique identity (id) and can transmit/receive random packets. A random packet means a payload of L symbols over F_2. Each packet has a unique identifier that consists of network id. We formulate the participating terminals as T_i, for $i = 0, ..., k$ in the setting of wireless communication networks where packet loss is quite common. There are two ways the terminals communicate: (1) *Broadcast*: a terminal T_i transmits a packet once. (2) *Reliable broadcast*: A terminal T_i ensures that all other terminals $T_{j \neq i}$ receives a packet that T_i sent out through acknowledgment and re-transmissions. Under the network formulation, the communication between T_0 and T_i, for $i = 1, ..., k$ experiences *packet erasures*. It is common in various forms of wireless communication protocols as investigated in the pilot study (Pacific Gas & Electric Company, 2018) due to the following reasons or a combination of them. (1) *Noise produced by interference or obstacles in the communication path* (2) *Reduced transmission power in the communication path* (3) *Low signal to noise ratio (SNR)* (4) *Collision because of concurrent transmission* (5) *Fading and multiple paths* (6) *Terminals experience some "offline" time periods*

If a packet transmitted in the network is missed by a terminal's radio receiver, that terminal experiences a packet erasure (misses packet contents). Specifically, we propose to turn the shortcomings of missing messages in the communication, such as inconsistent communication quality observed in wireless communications (Pacific Gas & Electric Company, 2018), into a building block to generate secret keys based on the information-theoretic secure scheme of secret generation (Siavoshani et al., 2011).

In Table 1, we explain the meaning of commonly used symbols throughout this section. Next, we describe the core of our key-establishment scheme based on the information-theoretic secure scheme from Siavoshani et al. (2011). The protocol enables the enterprise operator to

Table 1: Commonly Used Symbols

Symbol	Description
N	Number of x-packets sent by each terminal
T_i	ith transmit node.
k	Total number of nodes
$N_{T_i - T_j}$	Number of packets shared between terminal T_i and T_j.
T_{id}	Chosen Terminal id
TS	Time stamp
M_{T_i}	Number of y-packets constructed by a chosen terminal
$(M_i)_{T_{id}}$	Number of y-packets reconstructed by terminals other than the chosen ones

form secret keys with each of the IoT-HED without a centralized key server [1]. Since the terminals themselves randomly form Internet of IoT-HEDs among them, the communication path from the IoT data server terminal to a destination IoT-HED terminal is established and secured by the random graph theory (Eschenauer and Gligor, 2002). Therefore, the IoT-HED terminals and the IoT data server terminals can run the protocol to get the secret key for communication to receive control profiles from the data server periodically. The protocol for the secret key establishment is shown as follows, where T_i stands for a terminal of a IoT-HED and T_j stands for a terminal owned by the IoT data server or other neighboring terminals not owned by T_i.

4.2 Pairwise Key Establishment between the Utility and the IoT-HED

1. Each terminal T_i $(i = 1, 2..., k)$ broadcasts N packets denoted as x-packets.
2. Each terminal T_i $(i = 1, 2..., k)$ and other terminals $T_{j \neq i}$ $(j = 1, 2, , k)$ reliably broadcast the identities of the x-packets it received correctly[2]
3. Each terminal in the network creates a two-dimensional list maintaining the identities of packets it reliably broadcasts. Table 2 shows an example of a list of x-packets received among three terminal.
4. Each terminal reliably broadcasts $N_{T_i - T_j}$, the number of x-packets on its list of received x-packets.
5. T_i, who wants to generate a secret, identifies at least two terminals from among $T_{j \neq i}$ $(j = 1, 2, ..., k)$ with whom it wants to start generating linear combination of x-packets. Then T_i reliably broadcasts the list of its selected terminals.
   (a) The linear combination of x-packets are called y-packets.
   (b) The terminals T_j's are chosen to be the ones who have max($\forall ij N_{T_i - T_j}$) packets shared with T_i.
6. Each chosen terminal T_j constructs M_{T_i} y-packets by linear combinations of the $N_{T_i - T_j}$ x-packets it has. The linear combinations are explained in Section 5.
7. T_j's then reliably broadcast the identities of the x-packets on which the y-packets are constructed to all the terminals with the terminal identity T_{id}.
8. Each terminal T_i (other than T_j's) reconstructs as many as $(M_i)_{T_{id}}$ (empirically[3] calculated) y-packets as it can, using the $N_{T_i - T_j}$ x-packets it received.

[1] In fact, our protocol enables any pair of terminals T_i and T_j to create a shared secret S. The reason for reserving this capability is to allow device authentication. We will describe the authentication in another paper.

[2] The process continues for all the terminals in the network who receive control profiles from T_0.

[3] Probability of packet erasure is calculated on the basis of "number of packets missed by the terminals/total transmitted packets by that terminal".

Table 2: List of shared packets among terminals

	1	2	3	......
1	-	x11,x12,x14	x11,x13,x15,x16,x17	......
2	x21,x23	-	x22,x24,x25	......
3	x33,x34,x35	x31,x35	-	......
......	......	......	......	

5 SECURITY ANALYSIS

In this section, Information security proof of the protocol proposed in Section 4.2 is presented. Our contribution is formally writing the rigorous proofs which are in Information theoretic sense. Also, Unconditional Security of the proposed protocol is presented in Section 5.2 and in Section 5.3 we show that interaction proves lower bounds on the probability of guessing the packets interchanged by terminals (the detailed proofs to be presented in extended version due to space reason. Only theorems and example results are presented here).

5.1 Information Theoretic Analysis

In our proposed scheme, x-packets are shared by the terminals. The algorithm constructs the y-packets, z-packets and s-packets using x-packets shared by terminals. Let us call them X, Y, Z, S matrices that have as many rows as the x, y, z and s-packets respectively. In order to do the protocol analysis and give the information-theoretic proof of the scheme used in IoT-HED, we first give the mathematical model of the constructions used in the algorithm. From here on we will use the terms Alice and Bob for legitimate terminals T_i's and Eve for adversary.

5.1.1 Construction of y-packets

The sender T_0, or Alice, considers each subset of the recipient terminals T_i. She identifies the N_{T_i} X-packets that were received by each terminal T_i in the subset but no other terminals. She then creates M_{T_i} linear combinations $y_1, ..., y_{M_{T_i}}$ of these packets as $Y = A_y X$. Let each row of the matrix X comprise the x-packets received by a corresponding recipient terminal. The rows of the matrix Y are the M_{T_i} y-packets and A_y is the generator matrix of Maximum Distance Separable (MDS) linear code with parameters [N_{T_i}, M_{T_i}, N_{T_i}-M_{T_i+1}]. Each terminal T_i can reconstruct these y-packets using the MDS codes. Lemma 5.1, proves that this construction is information theoretic-secure against Eve.

Lemma 5.1. *Consider a set of N x-packets, denoted as $x_1, ..., x_N$, and assume Eve has a subset of size (N-M) of the x-packets. Construct M y-packets, denoted $y_1, ..., y_M$, as*

$$Y = AX \tag{1}$$

where matrix X has as rows the N x-packets, matrix Y has as rows the M y-packets, and A is the generator matrix of a Maximum Distance Separable (MDS) linear code with parameters [N, M, *N−M+1]. Then the M y-packets are information-theoretically secure against Eve, irrespective of which subset (of size N - M) of the x-packets Eve has.*

5.1.2 Construction of z-packets

Z-packets are constructed in the algorithm as linear combinations of the y-packets by any set of terminals using a linear code as

$$Z = A_z Y \tag{2}$$

Where A_z has M rows and is constructed using any standard basis extension method (Horn and Johnson, 2012). Each terminal $T_{i\neq 0}$ can combine M - M_i z-packets with the M_i y-packets it already has and reconstruct all the M y-packets. Choosing the z-packets can be done using standard network-coding techniques (Fragouli et al., 2007).

$$\text{rank}(A_z) = M - L \tag{3}$$

5.1.3 Construction of s-packets

Secret generating terminal constructs the L s-packets, denoted as $s_1, ..., s_L$ using a linear code, i.e.,

$$S = A_s Y \tag{4}$$

where A_s is constructed using any standard basis extension method so that

$$\text{rank} \begin{bmatrix} A_s \\ A_z \end{bmatrix} = M$$

Lemma 5.2 shows that the s-packets are secure from Eve.

Lemma 5.2. *Consider a set of M y-packets, denoted y_1,..., y_M, and a set of (M - L) z-packets, denoted z_1,..., z_{M-L}, related as Z = A_zY, where matrix Y has as rows the M y-packets, matrix Z has as rows the (M - L) z-packets, and A_z is a known (M - L) × M full rank matrix. Assume that Eve knows all the z-packets known as W. Using any standard basis-extension method, find an L × M matrix A_s, with rank (A_s) = L, such that*

$$rank \begin{bmatrix} A_s \\ A_z \end{bmatrix} = M$$

Then we can construct L s-packets, denoted s_1,..., s_L, as

$$S = A_s Y \tag{5}$$

where matrix S has as rows the s-packets. The construction satisfies the following results:

$$H(S|Z) = H(S) \tag{6}$$

$$H(S|W) = H(S) \tag{7}$$

$$I(S|Z) = 0 \tag{8}$$

$$I(S;Z|W) = 0 \tag{9}$$

5.2 Unconditionally Secure Secret Key Agreement

One natural assumption is that the random experiment generating the X, Y, and Z packets are repeated many times independently, as, in our protocol, all the terminals are producing the bits randomly and uniformly. The unconditionally secure secret-key agreement takes place in a scenario the same as our enclave where terminals have access to an insecure channel to which a passive eavesdropper Eve also has perfect access (Bennett et al., 1988, Maurer, 1993). In such a scenario, terminals know the correlated random variables x, y, and z respectively, which are distributed according to some joint probability distribution P_{xyz}. The legitimate terminals share no secret key initially but are assumed to know P_{xyz} that they can agree with empirically in the first face of the proposed protocol. The protocol and codes used by terminals are known to Eve. It was proved in Maurer (1993) that the size of the secret key S that can be generated by any protocol, not necessarily the one as ours, is upper bounded by

$$H(S) \leq \max(I(X;Y|Z);I(X;Y)) + I(S;CZ) \tag{10}$$

where C is the total bits exchanged between the terminals over the public channel. In other words, if I(S; CZ) must be negligible (which is the goal of our key agreement protocol), then terminals cannot generate a key that is longer than the mutual information between X and Y. Moreover, because if Eve revealed her random variable Z for free, this could only help Alice and Bob to generate a secret key. Therefore, the remaining mutual information between X and Y when given Z, I (X; Y|Z), is also an upper bound on H(S). Note that both I (X; Y|Z) < I (X; Y) or I (X; Y|Z) > I (X; Y) is possible. In order to be able to prove lower bounds on the achievable size of a key shared by terminals in secrecy we need to make more specific assumptions about the distribution P_{xyz}. One natural assumption is that the random experiment generating x, y, z is repeated many times independently as in case of our protocol; Alice, Bob and Eve receive $X^N = [x_1, x_2, ..., x_N], Y^N = [y_1, y_2, ..., y_N], Z^N = [z_1, z_2, ..., z_N]$, respectively, where

$$P_{X^N Y^N Z^N} = \prod_{i=1}^{N} P_{X_i} P_{Y_i} P_{Z_i} \tag{11}$$

5.3 Interaction is More Powerful than One-Way Transmission

It is demonstrated in this section that for certain probability distributions P_{xyz} or epsilon in case of our protocol, it is crucial for Alice and Bob to be able to use the public channel in both directions, possibly during several rounds. Regardless of Eve's probability ϵ_E compared to legitimate terminals, it is impossible for Eve to decode with any certainty (information). (Due to space concerns, We will present the detail proof in the extended version of this paper).

$$p_{correct} = (\delta_A \delta_B + \epsilon_A \epsilon_B)^N \tag{12}$$

The following result is the Eve's mutual information about the bit sent by the terminal.

$$I_E = \sum_{w=0}^{N} \binom{N}{w} \frac{p_w}{p_{accept}} (1 - h(\frac{p_w}{p_w + p_{N-w}})) \tag{13}$$

For sufficiently large N, we have $I_E < I_B$ and β arbitrarily small. By adding an appropriate number of such bits modulo 2, Eve's information about the resulting bit can be made arbitrarily small while at the same time keeping the probability that Alice's and Bob's bits disagree arbitrarily small. We summarize the simulation results in Section 6.

6 EXPERIMENTAL EVALUATIONS

Although various novel schemes have been developed but the practical application of these proposals seems unrealistic due to the limited processing ability and energy sources of sensors makes it possible for only basic and weak implementation of cryptographic algorithms. Results of Section 6.1 and 6.2, clearly shows that our scheme is suitable for any low cost IoT devices. Without using any central server or PKI scheme, nodes can safely negotiate an initial key.

We may imagine that several IoT nodes in an area where they are communicating with the same cluster router. The adversary may or may not be in the same cluster. Therefore, the secret generated by our scheme will not reveal the adversary to obtain the secret key for that cluster. This could apply to the smart home scenario, where security cameras form the cluster. It could also apply to flying ad hoc networks, where drones form a cluster and they do not need sessions keys from a centralized server but form the session key among themselves. As per our knowledge no such scheme to share a key between IoT devices exist in literature.

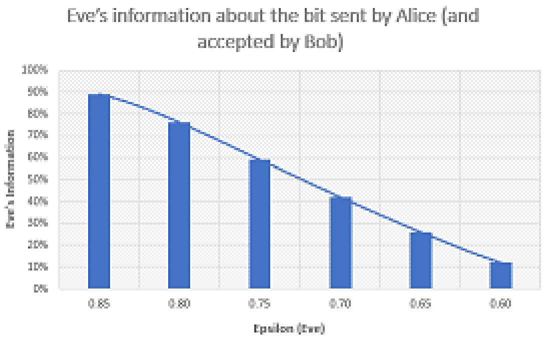


Figure 2: Eve's Information about the bits exchanged

Example 6.1. *Let ϵ_A = ϵ_B = 0.3, ϵ_E = 0.45 and N = 5. Then $p_{correct}$ = 0.0656357, p_{error} = 0.0130691, p_{accept} = 0.0787048, a_{00} = 0.31, a_{01} = 0.27, a_{10} = a_{11} = 0.21, p_0 = 0.00327133, p_1 = 0.00290192, p_2 = 0.00258017, p_3 = 0.00229995, p_4 = 0.00205588, and p_5 = 0.0018433. Hence, β = 16.6% compared to γ = 44.6% and thus Bob receives the selected bits much more reliably than Eve. One further obtains I_B = 0.351406 and I_E = 0.0118276, i.e., Eve's information about the bit sent by Alice (and accepted by Bob) is 97% smaller than Bob's information. Eve's information about the bit sent by Alice and accepted by Bob under various values of ϵ_E is shown in Figure 2. The simulation results showing Eve's average error probability of guessing the bits γ in percentage versus Terminal's bit-error probability in guessing the bits β are shown in Figure 3. These statistics validate the theoretical results proved above about the ill-information of malicious entity about the secret key, regardless of the adversary's computational capability.*

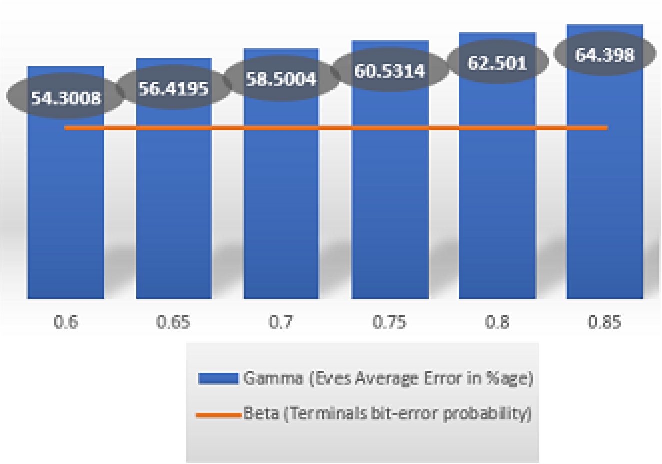


Figure 3: Eve's average error probability (guessing the bits) γ in percentage vs Terminal's bit-error probability (guessing the bits) β. Denote the epsilons for Terminal and Eve as ϵ_T and ϵ_E. In simulation, ϵ_E ranges from 0.6 to 0.85 where $\epsilon_T = 0.4$, $\beta = 40.13\%$.

Table 3: Different parameters used in the simulation. AODV, ad hoc on-demand distance vector

Parameter	**Description**
Platform Ubuntu 18.04 LTS	
Tool used (NS2) and NSG2.1	
Number of gateway nodes/edge nodes	1
Number of users or IoT device	10
Simulation time	600 s
Communication range of sensors/IoT devices	50m
Routing protocol	AODV

The simulation test-bed is discussed in Table 3 were executed on a workstation: Intel(R) Core(TM) i7-4500 1.80 GHz, 8 GB Ram. Our scheme can achieve thousands of secret bits per second under various erasure probabilities, as shown in Table 4.

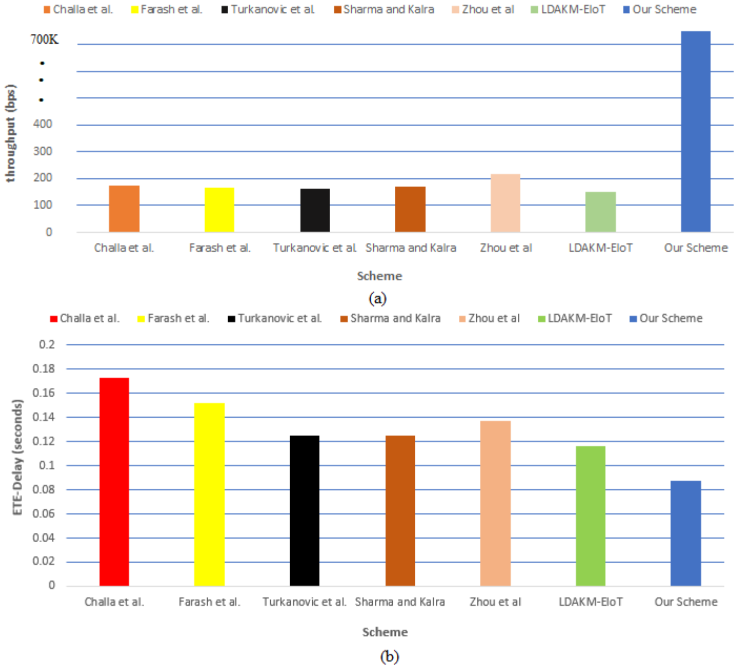
Table 4: Secret Generation Experiments Time (kbps) (ATP), Average End-to-End delay(ms) (AE2ED) and Packet delivery ratio (PDR)

No. of Nodes	**ATP[kbps]**	**AE2ED**	**PDR**
04 Nodes	781.49	202.47	95.88
06 Nodes	729.68	90.66	88.00
08 Nodes	792.67	81.54	89.18
10 Nodes	806.56	51.83	81.71

6.1 Comparative Analysis of the Network and End-to-End Delay

The network throughput can be calculated as “the number of bits transmitted per unit time, and it can be mathematically expressed as $(v_r \times |\rho|)/T_d$, where T_d is the total time (in seconds), ρ the size of a packet, and v_r the total number of received packets”. The simulation results in Figure 4(a) shows that our scheme has a better throughput rate (though our scheme don’t have the authentication phase) but still the throughput is approx. 700 times better which is a significant improvement in key establishment phase as compared to Challa et al. (2017), Farash et al. (2016), Turkanović et al. (2014), Sharma and Kalra (2019), Zhou et al. (2019) and Wazid et al. (2019).

The end-to-end delay (EED) is measured as “the average time taken by the data packets to arrive at a receiving node from a sender node, and it is mathematically expressed in the form $\Sigma(T_{rec_i} - T_{send_i})/v_p$, where T_{rec_i} and T_{send_i} are the receiving and sending time of a packet i, respectively, and v_p the total number of packets”. The simulation results in Figure 4(b) shows that our scheme has the least end to end delay as compared to Challa et al. (2017), Farash et al. (2016), Turkanović et al. (2014), Sharma and Kalra (2019), Zhou et al. (2019) and Wazid et al. (2019).



(a)

(b)

Figure 4: Comparison of Network Parameter

6.2 Reliability and Efficiency

The simulation results of Table 4 indicates that the secret generation rates are in thousand of bits per second. Such a size is easily obtainable in a neighborhood of IoT-HEDs, typically 4 to 10 IoT-HEDs in number. Each packet generated is of a size of 1500 bits. The above results indicate that the secret generation rate is very high, and the end-to-end average delay is very low. The results include the wireless transmission time, which varies significantly in practice depending on the specific physical layer medium. Therefore, simulation validates that the proposed scheme overcomes the challenges outlined in Section 3.

In the presence of an adversary who cannot receive all packets, the simulation results validate the secrecy rate and assumptions. In the worst-case scenario, even if the Eve overhears all the x-packets received by a terminal, the terminals can still construct a secret between them securely. As stated in the protocol and its proof, terminals take turns in the process of generating secrets and then exclusive-or the generated secrets by the selected terminals to get a final secret key. The security is similar to the one-time pad. Minimum reliability is achieved for $n = 8$ terminals, i.e., 1, which means Eve will not be able to construct any secret or even near to it. For $n = 6$, the reliability decreased to 0.15. This decrease in the result makes the probability of guessing the secret bit as $2^{-R} = 2^{-0.15} = 0.87$, which is quite high, but for the entire packet with 1500 bits this probability is $2^{-0.15*1500} \approx 0$. These results strongly suggest that it is reasonable to generate secretly thousands of bits per second.

7 CONCLUSION

In this paper, we have proposed an efficient distributed key establishment scheme for the Internet of Things (IoT) comprised of heterogeneous edge devices. The scheme aims to address a new security challenge in the forming of a sensitive communication network that has devices from different owners. The formed big IoT networks have a broad deployment from a smart community to smart grids. We have modeled adversaries to be external to the communications. Our proposed scheme uniquely forms IoT devices to be in an enclave with a communication terminal of the next level data server. The key properties of our scheme that differentiate it from prior theoretical work are that it is scalable to many enclaves with an arbitrary number of nodes, and has polynomial complexity for legitimate terminals on the IoT network. The proposed cryptographic scheme comes with proved information-theoretic security. When compared with other schemes, our scheme has high throughput and low end to end delay which makes it a perfect fit for IoT's. A future direction of the work is to investigate an efficient authentication mechanism of forming the enclaves of IoT devices.

REFERENCES

Abdallah, W., Boudriga, N., Kim, D., and An, S. (2015). An efficient and scalable key management mechanism for wireless sensor networks. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 480–493. IEEE.

Ahlswede, R. and Cai, N. (2006). Transmission, identification and common randomness capacities for wire-tape channels with secure feedback from the decoder. In *General Theory of Information Transfer and Combinatorics*, pages 258–275. Springer.

AlDairi, A. et al. (2017). Cyber security attacks on smart cities and associated mobile technologies. *Procedia Computer Science*, 109:1086–1091.

Ardestanizadeh, E., Franceschetti, M., Javidi, T., and Kim, Y.-H. (2009). Wiretap channel with secure rate-limited feedback. *IEEE Transactions on Information Theory*, 55(12):5353–5361.

Bassi, G., Piantanida, P., and Shitz, S. S. (2018). The wiretap channel with generalized feedback: Secure communication and key generation. *IEEE Transactions on Information Theory*, 65(4):2213–2233.

Baumgart, I., Borsig, M., Goerke, N., Hackenjos, T., Rill, J., and Wehmer, M. (2019). Who controls your energy? on the (in) security of residential battery energy storage systems. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE.

Bennett, C. H., Brassard, G., and Robert, J.-M. (1988). Privacy amplification by public discussion. *SIAM journal on Computing*, 17(2):210–229.

Cai, N. and Yeung, R. W. (2011). Secure network coding on a wiretap network. *IEEE Transactions on Information Theory*, 57(1):424435.

Challa, S., Wazid, M., Das, A. K., Kumar, N., Reddy, A. G., Yoon, E.-J., and Yoo, K.-Y. (2017). Secure signature-based authenticated key establishment scheme for future iot applications. *IEEE Access*, 5:3028–3043.

Cheng, C., Lu, R., Petzoldt, A., and Takagi, T. (2017). Securing the internet of things in a quantum world. *IEEE Communications Magazine*, 55(2):116–120.

Cheng, Q., Hsu, C., and Harn, L. (2020). Lightweight noninteractive membership authentication and group key establishment for wsns. *Mathematical Problems in Engineering*, 2020.

Choi, J., Shin, W., Kim, J., and Kim, K.-H. (2020). Random seed generation for iot key generation and key management system using blockchain. In *2020 International Conference on Information Networking (ICOIN)*, pages 663–665. IEEE.

Csiszár, I. and Narayan, P. (2008). Secrecy capacities for multiterminal channel models. *IEEE Transactions on Information Theory*, 54(6):2437–2452.

Csiszr, I. and Narayan, P. (2005). Secrecy capacities for multiterminal channel models. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, page 21382141. IEEE.

Dai, B. and Luo, Y. (2018). An improved feedback coding scheme for the wire-tap channel. *IEEE Transactions on Information Forensics and Security*, 14(1):262–271.

Del Pino, R., Lyubashevsky, V., Neven, G., and Seiler, G. (2017). Practical quantum-safe voting from lattices. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1565–1581.

Ekrem, E. and Ulukus, S. (2009). Secrecy capacity of a class of broadcast channels with an eavesdropper. *EURASIP Journal on Wireless Communications and Networking*, 2009:1.

Eschenauer, L. and Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM.

Farash, M. S., Turkanović, M., Kumari, S., and Hölbl, M. (2016). An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. *Ad Hoc Networks*, 36:152–176.

Fragouli, C., Soljanin, E., et al. (2007). Network coding fundamentals. *Foundations and Trends® in Networking*, 2(1):1–133.

Goldreich, O. (2006). *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press.

Harn, L. and Hsu, C.-F. (2015). Predistribution scheme for establishing group keys in wireless sensor networks. *IEEE Sensors Journal*, 15(9):5103–5108.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Hsiao, T.-C., Chen, T.-L., Chen, T.-S., and Chung, Y.-F. (2019). Elliptic curve cryptosystems-based date-constrained hierarchical key management scheme in internet of things. *Sensors and Materials*, 31(2):355–364.

Kanukurthi, B. and Reyzin, L. (2009). Key agreement from close secrets over unsecured channels. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, page 206223. Springer.

Khisti, A., Diggavi, S. N., and Wornell, G. W. (2012). Secret-key generation using correlated sources and channels. *IEEE Transactions on Information Theory*, 58(2):652–670.

Kumari, S., Das, A. K., Wazid, M., Li, X., Wu, F., Choo, K.-K. R., and Khan, M. K. (2017). On the design of a secure user authentication and key agreement scheme for wireless sensor networks. *Concurrency and Computation: Practice and Experience*, 29(23):e3930.

Lai, L. and Ho, S.-W. (2015). Key generation algorithms for pairwise independent networks based on graphical models. *IEEE Transactions on Information Theory*, 61(9):4828–4837.

Lai, L., Liang, Y., and Du, W. (2012). Cooperative key generation in wireless networks. *IEEE Journal on Selected Areas in Communications*, 30(8):1578–1588.

Lai, L., Liang, Y., and Poor, H. V. (2011). A unified framework for key agreement over wireless fading channels. *IEEE Transactions on Information Forensics and Security*, 7(2):480–490.

Lee, A. (2013). Electric sector failure scenarios and impact analyses. *National Electric Sector Cybersecurity Organization Resource (NESCOR) Technical Working Group*, 1.

Maurer, U. M. (1993). Secret key agreement by public discussion from common information. *IEEE transactions on information theory*, 39(3):733742.

Mirvaziri, H. and Hosseini, R. (2020). A novel method for key establishment based on symmetric cryptography in hierarchical wireless sensor networks. *Wireless Personal Communications*, pages 1–19.

Mosca, M. (2018). Cybersecurity in an era with quantum computers: will we be ready? *IEEE Security & Privacy*, 16(5):38–41.

Nicanfar, H., Jokar, P., Beznosov, K., and Leung, V. C. (2014). Efficient authentication and key management mechanisms for smart grid communications. *IEEE systems journal*, 8(2):629640.

Nitinawarat, S., Ye, C., Barg, A., Narayan, P., and Reznik, A. (2010). Secret key generation for a pairwise independent network model. *IEEE Transactions on Information Theory*, 56(12):6482–6489.

Pacific Gas & Electric Company, P. (2018). Epic 2.03a: Test capabilities of customer-sited behind-the-meter smart inverters. `https://www.pge.com/pge_global/common/pdfs/about-pge/environment/what-we-are-doing/electric-program-investment-charge/PGE-EPIC-Project-2.03a.pdf?WT.mc_id=Vanity_epicinterimreport-SmartInverters`. Accessed August 7, 2018.

Prabhakaran, V. M., Eswaran, K., and Ramchandran, K. (2012). Secrecy via sources and channels. *IEEE transactions on information theory*, 58(11):6747–6765.

Salimi, S. and Skoglund, M. (2012). Secret key agreement using correlated sources over the generalized multiple access channel. In *2012 IEEE Information Theory Workshop*, pages 467–471. IEEE.

Salimi, S., Skoglund, M., Golic, J. D., Salmasizadeh, M., and Aref, M. R. (2013). Key agreement over a generalized multiple access channel using noiseless and noisy feedback. *IEEE Journal on Selected Areas in Communications*, 31(9):1765–1778.

Sharma, G. and Kalra, S. (2019). A lightweight user authentication scheme for cloud-iot based healthcare services. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 43(1):619–636.

Siavoshani, M. J., Pulleti, U., Atsan, E., Safaka, I., Fragoulia, C., Argyraki, K., and Diggavi, S. (2011). Exchanging secrets without using cryptography. *arXiv preprint arXiv:1105.4991 v1.*

Tavangaran, N., Schaefer, R. F., Poor, H. V., and Boche, H. (2018). Secret-key generation and convexity of the rate region using infinite compound sources. *IEEE Transactions on Information Forensics and Security*, 13(8):2075–2086.

Tu, W., Goldenbaum, M., Lai, L., and Poor, H. V. (2016). On simultaneously generating multiple keys in a joint source-channel model. *IEEE Transactions on Information Forensics and Security*, 12(2):298–308.

Turkanović, M., Brumen, B., and Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, 20:96–112.

Wang, N., Zhang, N., and Gulliver, T. A. (2013). Cooperative key agreement for wireless networking: Key rates and practical protocol design. *IEEE Transactions on Information Forensics and Security*, 9(2):272–284.

Wazid, M., Das, A. K., Shetty, S., JPC Rodrigues, J., and Park, Y. (2019). Ldakm-eiot: Lightweight device authentication and key management mechanism for edge-based iot deployment. *Sensors*, 19(24):5539.

Wyner, A. D. (1975). The wire-tap channel. *Bell system technical journal*, 54(8):13551387.

Xu, P., Cumanan, K., Ding, Z., Dai, X., and Leung, K. K. (2016a). Group secret key generation in wireless networks: algorithms and rate optimization. *IEEE Transactions on Information Forensics and Security*, 11(8):1831–1846.

Xu, P., Ding, Z., Dai, X., and Karagiannidis, G. K. (2016b). On the private key capacity of the m-relay pairwise independent network. *IEEE Transactions on Information Theory*, 62(7):3831–3843.

Xu, P., Ding, Z., Dai, X., and Karagiannidis, G. K. (2016c). Simultaneously generating secret and private keys in a cooperative pairwise-independent network. *IEEE Transactions on Information Forensics and Security*, 11(6):1139–1150.

Ye, C. and Reznik, A. (2007). Group secret key generation algorithms. In *2007 IEEE International Symposium on Information Theory*, pages 2596–2600. IEEE.

Zhang, H., Lai, L., Liang, Y., and Wang, H. (2014). The capacity region of the source-type model for secret key and private key generation. *IEEE Transactions on Information Theory*, 60(10):6389–6398.

Zhang, H., Liang, Y., Lai, L., and Shitz, S. S. (2017). Multi-key generation over a cellular model with a helper. *IEEE Transactions on Information Theory*, 63(6):3804–3822.

Zhang, J., Li, H., and Li, J. (2020). An improved key pre-distribution scheme based on the security level classification of keys for wireless sensor networks. *International Journal of Information and Computer Security*, 12(1):40–52.

Zhou, H., Huie, L. M., and Lai, L. (2014). Secret key generation in the two-way relay channel with active attackers. *IEEE Transactions on Information Forensics and Security*, 9(3):476–488.

Zhou, L., Li, X., Yeh, K.-H., Su, C., and Chiu, W. (2019). Lightweight iot-based authentication scheme in cloud computing circumstance. *Future Generation Computer Systems*, 91:244–251.

Group Identity-Based Identification: Definitions and Construction

Apurva Kiran Vangujar *[1], **Tiong-Sik Ng**[2], **Ji-Jian Chin**[1], and **Sook-Chin Yip**[1]

[1]*Faculty of Engineering, Multimedia University*
[2]*School of Electrical & Electronic Engineering, Yonsei University, Seoul*

E-mail: apurva710@gmail.com
**Corresponding author*

ABSTRACT

As an extension of identification schemes in multiparty setting, we propose the first definitions and construction for a Group Identity-Based Identification (Group-IBI) scheme. The Group-IBI involves a group manager ($\mathcal{GM}$) that is in charge of a specific group, which in turn manages several group members. The $\mathcal{GM}$'s role is not only to control the registration and revocation of the members, but also to perform an identification protocol with a verifier as a whole entity, i.e., a group. The Group-IBI scheme that we proposed is potentially suitable for numerous real-world online applications such as e-shopping, e-banking, and e-voting where consensus of all members of a group is required to be derived before proceeding with authentication. In this paper, we propose the first definitions and security models for Group-IBI. We also show the first provable-secure construction that is pairing free by using the Schnorr identity-based identification (IBI) and Schnorr signature.

Keywords: Multiparty schemes, identity-based identification scheme, provable security.

1 INTRODUCTION

Chaum and Pedersen (1992) introduced the group signature scheme (GSS), an extended version of a generic signature scheme. The entities involved in the GSS include a group member, several group members, and an adversary group member. In the GSS, the group members are only able to mark messages anonymously on behalf of the group. Since the signatures are verified by a group public key instead of an individual public key, the identity of the signer is not disclosed, thus maintaining the anonymity of the signer.

In a practical setting, GSS can be used by employees to sign documents as a representative, or an employee of the company. However, since it is not secure to store the messages, the verifier has to be aware that the group public key is authentically from the group itself. A few years later, Lysyanskaya and Ramzan (1998) combined the GSS with blind signatures for the application of electronic cash. In the proposed scheme, the setting of multiple banks is possible. To maintain several layers of anonymity, the identity of the spender is anonymous to the bank while the spenders bank is not visible to the vendor as well.

In some works related to the GSS such as the ones by Chen and Pedersen (1994), Camenisch and Stadler (1997), Camenisch and Michels (1998), Camenisch and Michels (1999), and Kiayias and Yung (2005), the properties of the GSS were elaborated in depth. However, some GSS are said to be restricted and inefficient. As an example, one major challenge faced by the GSS is to be able to maintain large numbers of group signatures. Another related work would be the one by Bellare et al. (2003), where theoretical foundations for the group signature primitive were provided. In addition to that, Bellare et al. also constructed the GSS based on general assumptions.

Though there has not been much progress for some period of time, GSS schemes began to resurface in 2010, such as the works done by Gordon et al. (2010). In particular, Gordon et al.'s work is a lattice version of the GSS using the learning with errors (LWE) assumption under the random oracle model. Then, Langlois et al. (2014) proposed another lattice version of the GSS, where the revocation is done locally during the verification. Considering that all the users in the GSS are associated to a group, the security of the users are affected once the group's security is compromised. Hence, we propose the Group-IBI scheme as a solution, to protect the security of the users in a group even if the group is compromised.

1.1 Motivations and Contributions

Our contribution in this work is two-fold. The first is to formalize and define the security notions and definitions for Group-IBI. The second is we show a concrete construction using Schnorr IBI (Tan et al., 2011) and signatures (Schnorr, 1989). We describe our motivations for Group-IBI as follow.

The security of most company systems is dependent on a specialized network administration that is specific to the company itself. While it is important to have an internal management to track any ongoing transactions, it is also a key property to be able to represent the company as a whole for any external transactions to prove that the transaction is authentically done by the company instead of an impostor. The proposed Group-IBI is able to handle this use-case, which is one of its significant properties.

With the consideration that a group has to be handled internally within by the $\mathcal{GM}$ and the group members, a trusted authority ($\mathcal{TA}$) is utilized to generate the group key pairs for many groups, where the group keys will be distributed to each group manager. However, each user key pair is generated by the $\mathcal{GM}$ himself without the interference of the $\mathcal{TA}$ or any other parties to ensure that the members management stays within the group itself. The $\mathcal{TA}$ stores all the key pairs which is generated by the $\mathcal{GM}$. It is significant to have the $\mathcal{TA}$ to maintain the whole group security.

For example, consider an e-voting system in a parliament consisting of a few parties and a session chair, the Group-IBI aims to tackle this problem by capturing the respective party in a group. Each party consisting of party members is handled by a $\mathcal{GM}$ that is in charge of registering and revoking the members of a specific party. The $\mathcal{GM}$ is also in charge of obtaining the consents from all the party members before performing a transaction. Particularly, the $\mathcal{GM}$ performs a group identification to represent the party with a verifier: the session chair.

Given that the signature proposed by Schnorr (1989) is one of the de-facto digital signature (DS) schemes to date, the Group-IBI scheme is initialized using variants of the Schnorr IBI and Schnorr DS after a general form of the Group-IBI is proposed. Tan et al. (2011) and Katz and Wang (2003)'s works are used for the Schnorr IBI and Schnorr DS respectively, where a security proof is provided after the application. In the security proof for the Group-IBI, we show that the security of the Group-IBI is tied to the scheme that is initialized with, in this case the Schnorr variants.

1.2 Organization

The paper is organized as follows. Section 2 begins with some preliminaries including assumptions and security models. In Section 3, we propose the Group-IBI scheme in a general form. The security proof and model for the Group-IBI scheme is then presented in Section 4. The application of the Group-IBI scheme using Schnorr variants is elaborated in Section 5. Finally we conclude with some discussions and future work in Section 6.

2 DEFINITIONS

In this section, we present the definition of Identity-Based Identification (IBI) and the Digital Signature (DS) schemes, where the Group-IBI is viewed as a combination of both schemes. In consideration that the Group-IBI involves a multiparty setting, we will only present the security model after we have defined the Group-IBI scheme in Section 4.

2.1 Identity-Based Identification

The identification scheme was initially proposed by Fiat and Shamir (1986). Boneh and Franklin (2003) pioneered the identity-based encryption scheme that led to the flourishing of identity-based cryptography. In later years, Bellare et al. (2009) constructed a more secure IBI scheme and constructed based on the zero-knowledge proof that results in higher efficiency.

In recent years, some advances were made in the field of IBI with work such as Barapatre and Rangan (2013),Vangujar et al. (2019) and Chia and Chin (2020), which cover IBI schemes from key encapsulation mechanisms, hierarchical IBI and tighter proofs for IBI schemes respectively.

The definition of an Identity-Based Identification (IBI) from Kurosawa and Heng (2004) is presented as follows:

Definition 2.1. *An identity-based identification scheme $IBI = (\mathcal{S}, \mathcal{E}, (\mathcal{P}, \mathcal{V}))$ consists of three polynomial-time algorithms: Setup, Extract, and Identification. The algorithms are described as follows:*

1. ***Setup.*** *$(\mathcal{S})$: Given the security parameter 1^k as an input, a pair of master public and secret keys (m_{pk}, m_{sk}) is generated. m_{pk} is known to the public but m_{sk} is only made known to the public key generator (PKG), or also known as the $\mathcal{TA}$.*
2. ***Extract.*** *$(\mathcal{E})$: $\mathcal{TA}$ takes in a public identity ID and (m_{pk}, m_{sk}) as inputs to generate a corresponding user secret key, u_{sk}.*
3. ***Identification Protocol.*** *$(\mathcal{P}, \mathcal{V})$: Using (m_{pk}, u_{sk}, ID) and (m_{pk}, ID) as inputs for $\mathcal{P}$ and $\mathcal{V}$ respectively, both parties will interact in a protocol as follows.*
   (a) ***commit*** (CMT)*: $\mathcal{P}$ sends* CMT *to $\mathcal{V}$.*
   (b) ***challenge*** (CHA)*: $\mathcal{V}$ responds $\mathcal{P}$ with a challenge* CHA.
   (c) ***response*** (RSP)*: $\mathcal{P}$ returns a response* RSP *to $\mathcal{V}$.*

   At the end of the protocol, $\mathcal{V}$ decides to accept or reject $\mathcal{P}$'s RSP *with a Boolean decision (I/O). A legitimate $\mathcal{P}$ should always be accepted.*

2.2 Digital Signatures

The definition of a Digital Signature (DS) by Kurosawa and Heng (2004) is presented.

Definition 2.2. *A digital signature scheme $DS = (\mathcal{KG}, \mathcal{SN}, \mathcal{VR})$ consists of three polynomial-time algorithms: Key Generation, Signing, and Verification. The algorithms are described as follows:*

1. ***Key Generation.*** *$(\mathcal{KG})$: A pair of public and secret keys are generated based on the security parameter input 1^k. The public key pk can be aired on an open channel, while the secret key sk is kept secret by the user.*
2. ***Signing.*** *$(\mathcal{SN})$: The user uses the secret key sk to sign on a message m to generate a signature, which is denoted as σ.*
3. ***Verification.*** *$(\mathcal{VR})$: The verifier takes the public key pk and σ as the input to ensure that the signature is genuinely signed by the user. If the signature is authentic, the algorithm returns "I", and "O" otherwise.*

2.3 Mathematical Assumptions

DDH assumption is defined from Boneh (1998).

Definition 2.3. *Decisional Diffie-Hellman Assumption (DDH). A Challenger $\mathcal{C}$ is said to (t, ε)-solve the DDH assumption if $\mathcal{C}$ runs in time at most t and furthermore:*

$$|\Pr[a, b, c \leftarrow \mathbb{Z}_q : \mathcal{C}(g, g^a, g^b, g^c) = 1] - \Pr[a, b \leftarrow \mathbb{Z}_q : \mathcal{C}(g, g^a, g^b, g^{ab}) = 1]| \geq \varepsilon$$

We say that the DDH assumption is (t, ε)-hard if no algorithm (t, ε)-solves the DDH assumption.

3 GROUP IDENTITY-BASED IDENTIFICATION

In this section, we present the Group Identity-Based Identification (Group-IBI) scheme in a general form alongside the security model for our scheme. Our proposed scheme may be viewed as a general framework for Group-IBI, where we will show an example instantiation in Section 5. The scheme is defined as transactions between $\mathcal{TA}$, $\mathcal{GM}$, and several group members.

1. **Setup** $(\mathcal{S})$**.** $\mathcal{TA}$ generates master key pairs (m_{pk}, m_{sk}).

2. **Extract** $(\mathcal{E})$**.** Phase 1 is run by the $\mathcal{TA}$, whereas Phases 2 and 3 are run by the $\mathcal{GM}$
   Phase 1. Using (m_{pk}, m_{sk}), $\mathcal{TA}$ generates group key pairs (g_{pk}, g_{sk}) and passes them to $\mathcal{GM}$.
   Phase 2. For each member of a group $(G_1, G_2, ..., G_n)$, a member is required to have an ID. To register as a member of the group, a group member sends ID to $\mathcal{GM}$. Using (g_{pk}, g_{sk}), $\mathcal{GM}$ generates user keys for member of the ID, (u_{pk}, u_{sk}).
   Phase 3. Then, $\mathcal{GM}$ sends the keys (u_{pk}, u_{sk}) to the group member. At the same time, $\mathcal{GM}$ stores (ID, u_{pk}) that is associated with the group member.

3. **Identification** $(\mathcal{P}, \mathcal{V})$**.**
   Phase 1. Suppose a group member wants to perform verification protocol as a group (i.e. G_1 wants to verify as a group). Using u_{sk}, G_1 generates a signature σ_1. G_1 notifies $\mathcal{GM}$ for a request to verify, and sends (u_{pk}, σ_1, ID) to $\mathcal{GM}$.
   Phase 2. $\mathcal{GM}$ then checks if the signature σ_1 is valid and verifies if the associated ID is within the list of members. If G_1 is a valid member in the list, $\mathcal{GM}$ then issues a notice to all other members in the group to generate their signatures and attach their u_{pk} as well. As $\mathcal{GM}$ receives the values for each members, $\mathcal{GM}$ also checks if the provided values are valid.
   Phase 3. Once all the values are obtained and are valid [1], $\mathcal{GM}$ then performs verification with a verification party $\mathcal{V}$, by attaching a signature generated from g_{sk}, σ_g as a representation of the group verification. $\mathcal{GM}$ then carries out the protocols CMT, CHA, and RSP with $\mathcal{V}$.
   Note: To revoke the membership of a group member, the $\mathcal{GM}$ only has to remove the associated ID and u_{pk} from the members list. Therefore, whenever a verification is done by a non-member, $\mathcal{GM}$ verifies if the produced signature is part of the member list and does not take into account if the signature is produced by a non-member for group verification.

[1] This means that consent from all members in the group are required to be able to perform a group verification.

4 SECURITY MODEL AND SECURITY PROOF

In this section, we cover two areas, namely the security model of the Group-IBI, and also the security proofs of impersonating a $\mathcal{GM}$ and a group member. We first define the security model of the Group-IBI, with reference to each of the security proofs described in Section 4.4.

4.1 Malicious third party

A malicious third party only has the capability to eavesdrop on messages and may try to impersonate other parties using the information obtained from eavesdropping to perform malicious activities on the group.

4.1.1 Impersonating $\mathcal{GM}$

Definition 4.1. *A malicious third party may try to impersonate a $\mathcal{GM}$ to perform a group verification. However, it is not possible if he does not have (g_{pk}, g_{sk}) which is tied to the (u_{pk}, u_{sk}) from the said group. Besides that, he does not have access to the members list within the group.*

4.1.2 Impersonating a group member

Definition 4.2. *A malicious third party may try to impersonate a group member to be part of the group (i.e., to be able to participate in activities as a legitimate part of the group). However, since a group member is required to register via the $\mathcal{GM}$, the malicious third party who tries to impersonate the group member may fail the member list checking if he is not able to produce σ that is generated from u_{sk}. Once he tries to register as a part of the group member via the $\mathcal{GM}$, he would fall into the case of a malicious group member.*

4.2 Malicious group member

Definition 4.3. *A malicious group member may try to replicate the role of the $\mathcal{GM}$ by generating his own (g_{pk}, g_{sk}) to target certain members within the group and trick them into giving him their consent to be able to perform group verification.*

4.2.1 Impersonating $\mathcal{GM}$

Definition 4.4. *A malicious group member impersonates the$\mathcal{GM}$ to perform group verification by himself without needing the consent of any group members or even the $\mathcal{GM}$. The security proof will be described in Section 4.4.2 to avoid impersonation of the$\mathcal{GM}$.*

4.2.2 Impersonating another group member

Definition 4.5. *The malicious group member may try to replicate another group member using their (ID, u_{pk}). The security proof for a malicious group member impersonating another group member is described in Section 4.4.1.*

4.3 Malicious $\mathcal{GM}$

The $\mathcal{GM}$'s task is to generate user keys (u_{pk}, u_{sk}) for group members using the group keys (g_{pk}, g_{sk}), while keeping track of the group members in a list. Besides that, the $\mathcal{GM}$ is also able to perform verification as a group representative. Considering the authority and role that a $\mathcal{GM}$ has over his own group, a malicious $\mathcal{GM}$'s goal is to be able to impersonate another group's $\mathcal{GM}$. With that in mind, the security proof follows the one described in Section 4.4.2.

4.4 Security Proof

4.4.1 Security Against Impersonation as Another Group Member

We define the security proof against impersonation as another group member, where a simulation game between a Challenger $\mathcal{C}$ and an Impersonator $\mathcal{I}$ is constructed. The goals of $\mathcal{C}$ and $\mathcal{I}$ are defined to solve the hard problem of the scheme and to impersonate as a member of the group, respectively.

Theorem 4.1. *The Group-IBI scheme above is $(t_{\text{Group-IBI}}, q_e, \varepsilon_{\text{Group-IBI}})$-secure against impersonation in the random oracle model if the hard problem of the signature holds, such that:*

$$\varepsilon_{\text{Group-IBI}} \approx \varepsilon_{\text{Sign}}$$
$$t_{\text{Group-IBI}} \approx \mathcal{O}(t_{\text{Sign}})$$

Proof. In this game, we construct a Challenger $\mathcal{C}$ making use of an Impersonator $\mathcal{I}$.
Phase 1
Setup. $\mathcal{C}$ obtains master public key, m_{pk}.
Extract Query. For an extract query of ID queried by $\mathcal{I}$, $\mathcal{C}$ computes and sends $(u_{pk\mathcal{I}}, u_{sk\mathcal{I}})$ to $G_{\mathcal{I}}$.
Identification Query. For an identification query on ID queried by $\mathcal{I}$, $\mathcal{C}$ checks if ID has been queried an extract query before. If so, $\mathcal{C}$ uses the existing $(u_{pk\mathcal{I}}, u_{sk\mathcal{I}})$ to return a valid transcript/conversation for $\mathcal{I}$; else, $\mathcal{C}$ runs extract query algorithm to generate $(u_{pk\mathcal{I}}, u_{sk\mathcal{I}})$ and returns a valid transcript/conversation to $\mathcal{I}$. The transcript may be a well-formed conversation created by $\mathcal{C}$ alone if it is a passive attack, or it may be a full conversation with $\mathcal{I}$ as a prover while $\mathcal{I}$ acts as a cheating verifier.

Phase 2
$\mathcal{I}$ pretends to be a valid group member using ID^*, where ID^* was queried during the extract query. $\mathcal{I}$ generates a signature $\sigma_{\mathcal{I}}$ and then sends the signature to $\mathcal{C}$. After $\mathcal{C}$ obtains the signature, $\mathcal{C}$ checks the validity of the signature [2]. If the signature produced by $\mathcal{I}$ is not valid, $\mathcal{C}$ aborts and it fails in the security game. Else, $\mathcal{C}$ can use the forgery $\sigma_{\mathcal{I}}$ to solve the hard problem used in the scheme and wins in the security game.

We now analyze the probability of aborts during the whole simulation process.

$$\begin{aligned}\Pr[\mathcal{C}\text{ wins}] &= \Pr[\mathcal{C}\text{ accepts }\sigma_{\mathcal{I}}] - \Pr[\mathcal{C}\text{ no abort}]\\ &\approx \varepsilon_{\text{Sign}} - 0\\ &\approx \varepsilon_{\text{Sign}}\end{aligned}$$

It is noted that during the query phase, the probability of aborts occurring is highly dependent on the signature scheme used. However, the abort that may occur during the query phase due to a hash collision is negligible. Therefore, it can be conjectured that if $\mathcal{I}$ is able to come up with a valid $\sigma_{\mathcal{I}}$, $\mathcal{I}$ has broken the signature scheme used in the Group-IBI.

It is noted that $\mathcal{O}(t_{\text{Sign}})$ is the time needed to query to the oracle. □

4.4.2 Security Against Impersonation as Group Manager

We define the security proof against impersonation as a group manager, where a simulation game between a Challenger $\mathcal{C}$ and an Impersonator $\mathcal{I}$ is constructed. The goals of $\mathcal{C}$ and $\mathcal{I}$ are defined to solve the hard problem of the scheme and to impersonate as a group manager, respectively.

Theorem 4.2. *The Group-IBI scheme above is $(t_{\text{Group-IBI}}, q_e, \varepsilon_{\text{Group-IBI}})$-secure against impersonation in the random oracle model if the hard problem of the IBI scheme used holds, such that:*

$$\begin{aligned}\varepsilon_{\text{Group-IBI}} &\approx \varepsilon_{\text{IBI}}\\ t_{\text{Group-IBI}} &\approx \mathcal{O}(t_{\text{IBI}})\end{aligned}$$

Proof. In this game, we construct a Challenger $\mathcal{C}$ making use of an Impersonator $\mathcal{I}$.
Phase 1
Setup. Similar to the proof described in Section 4.4.1.
Extract Query. Similar to the proof described in Section 4.4.1.
Identification Query. For an identification on a query ID by $\mathcal{I}$, $\mathcal{C}$ checks if ID is queried before during the extract query. If ID was queried before to $\mathcal{C}$, $\mathcal{C}$ uses the existing $(u_{pk\mathcal{I}}, u_{sk\mathcal{I}})$ to return as a valid transcript/conversation; else $\mathcal{C}$ runs extract query and then plays the role of a prover and performs the identification protocol with $\mathcal{I}$.

[2] It is noted that $\mathcal{I}$ has to produce the ID of a valid member in the group, else ID^* will fail when $\mathcal{C}$ does cross-checking on the validity of ID^* as a group member.

Phase 2
$\mathcal{I}$ pretends to be a valid identity ID^*. With the transcript/conversation produced by $\mathcal{I}$, $\mathcal{C}$ wins if the transcript/conversation is valid; else $\mathcal{C}$ aborts and loses the security game. *Note:* It is noted that the security proof for impersonation as $\mathcal{GM}$ is exactly the security proof for a generic IBI scheme. Therefore, we link the security proof for impersonation as $\mathcal{GM}$ to the IBI scheme that is used, which gives us Theorem 4.2, as required. □

5 APPLICATION OF GROUP-IBI

5.1 Group-IBI with Schnorr IBI and Schnorr DS

We initialize the Group-IBI using the Schnorr IBI and Schnorr DS. The Group-IBI involves the $\mathcal{TA}$ and the $\mathcal{GM}$ to perform a collective verification as a group, whereas the Schnorr signature involves the transaction between the $\mathcal{GM}$ and the group members for the members to prove their identity as a valid group member. We refer to Tan et al. (2011)'s tight Schnorr IBI variant and Katz and Wang (2003)'s tight Schnorr signature variant to instantiate the Group-IBI, with consideration that both schemes use the same hard problems and have the same key generation algorithms.

1. **Setup** $(\mathcal{S})$**.** On a security level 1^k, $\mathcal{TA}$ generates two large primes p and q, such that $q|(p-1)$. $\mathcal{TA}$ also generates $x \xleftarrow{R} \mathbb{Z}_q$ to compute $y_1 = g^{-x}$ and $y_2 = h^{-x}$ where $g, h \xleftarrow{R} \mathbb{G}$. Compute a hash function $H : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_q$. The master public key (m_{pk}) is $(p, q, g, h, y_1, y_2, H)$ while the master secret key (m_{sk}) is x.

2. **Extract** $(\mathcal{E})$**.**
   Phase 1. Given a group ID g_{ID}, selects a random integer $t \xleftarrow{R} \mathbb{Z}_q$. Then, $\mathcal{TA}$ computes $A = g^t, B = h^t$, and $s = t + x\alpha$ where $\alpha = H(g_{ID}, A, B, y_1, y_2)$. $\mathcal{TA}$ passes the group public keys $g_{pk} = (g_{ID}, g, h, y_1, y_2)$ and group secret keys $g_{sk} = (\alpha, s)$ to $\mathcal{GM}$.
   Phase 2. Consider a group member G_1 wants to register as a group member, he sends ID_1 to $\mathcal{GM}$. $\mathcal{GM}$ generates a random integer $a_1 \xleftarrow{R} \mathbb{Z}_q$ and then computes $y_{1,1} = g^{a_1}$ and $y_{2,1} = h^{a_1}$. In addition, $\mathcal{GM}$ also computes a hash function $H : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_q$
   Phase 3. $\mathcal{GM}$ passes the (u_{pk_1}, u_{sk1}) to G_1 as $((H, g, h, y_{1,1}, y_{2,1}), a_1)$. At the same time, $\mathcal{GM}$ stores $u_{pk_1} = (g, h, y_{1,1}, y_{2,1})$ that is associated with the ID of group member G_1, ID_1 in a list of members.

3. **Identification** $(\mathcal{P}, \mathcal{V})$**.**
   Phase 1. Suppose a group member wants to perform verification protocol as a group (i.e. G_1 wants to verify as a group). G_1 generates a random salt $r_1 \xleftarrow{R} \mathbb{Z}_q$ and computes $A_1 = g^{r_1}, B_1 = h^{r_1}, c_1 = H(PK, A_1, B_1, m_1), s_1 = a_1c_1 + r_1 \pmod q$. G_1 then sends the generated signature $\sigma_1 = (c_1, s_1)$ alongside the message m_1 [3] to the $\mathcal{GM}$.
   Phase 2. To check the validity of the signature σ_1, $\mathcal{GM}$ retrieves u_{pk_1} from the members

[3] G_1 may enclose the request for transaction (i.e. tr_1) alongside his ID ID_1 for verification as the message $m_1 = (tr_1||ID_1)$ G_1 may also enclose ID_1 separate from the message to notify $\mathcal{GM}$ that he is the member that is requesting for a group verification transaction.

list and computes $A'_1 = g^{s_1} y_{1,1}$ and $B'_1 = h^{s_1} y_{2,1}$. If the value of $H(PK, A'_1, B'_1, m_1) = c_1$, the signature is authentic and is a valid signature from the group member, since he is able to retrieve u_{pk_1} from the members list. $\mathcal{GM}$ then issues a notice to all other members in the group to generate their signatures with the transaction details (format of the message, such as $m_n = (tr_1 || ID_n)$). As $\mathcal{GM}$ receives the values for each members, $\mathcal{GM}$ also checks if the provided signatures are valid. Once all the values are obtained and are valid, $\mathcal{GM}$ is then able to perform verification with a verification party as a group representative.
Phase 3. $\mathcal{GM}$ performs the transaction with a verifier $\mathcal{VR}$. The verification protocol are carried out as follows.

(a) CMT : $\mathcal{GM}$ computes $A = g^s {y_1}^\alpha$ and $B = h^s {y_2}^\alpha$. Then, $\mathcal{GM}$ generates a random salt $r \xleftarrow{R} \mathbb{Z}_q$, computes $X = g^r$ and then sends (A, B, X) to $\mathcal{VR}$.

(b) CHA : $\mathcal{VR}$ then generates a challenge $c \xleftarrow{R} \mathbb{Z}_q$ and sends c to $\mathcal{GM}$.

(c) RSP : $\mathcal{GM}$ computes the response value $y = r + cs \pmod q$ and then sends the value of y to $\mathcal{VR}$.

$\mathcal{VR}$ accepts the value of if and only if the value of $g^y = X \cdot (A / {y_1}^\alpha)^c$, where $\alpha = H(g_{ID}, A, B, y_1, y_2)$.

5.2 Security Proof

5.2.1 Security Against Impersonation as Another Group Member

In this section, we present a full security proof for the Schnorr instantiated Group-IBI.

Theorem 5.1. *The Schnorr Group-IBI scheme above is* $(t_{\text{Group-IBI}}, q_e, \varepsilon_{\text{Group-IBI}})$*-secure against impersonation in the random oracle model if the Decisional Diffie-Hellman (DDH) problem holds, such that:*

$$\varepsilon_{\text{Group-IBI}} \geqslant \varepsilon_{\text{DDH}} + 2(q_e + 1)q^{-1}$$
$$t_{\text{Group-IBI}} \geqslant t_{\text{DDH}} + 2.4(q_e + 1)t_{exp}$$

where q_e *is the total extract queries that are queried by an impersonator* $\mathcal{I}$ *and assuming a two-exponent multi-exponentiation takes time* $1.2t_{exp}$.

Proof. In this game, we construct Challenger $\mathcal{C}$ making use of an Impersonator $\mathcal{I}$ in the Schnorr Group-IBI environment.
Phase 1
Setup. $\mathcal{C}$ sets master public key, m_{pk} as $(p, q, g, h, y_1, y_2, H)$.
Extract Query. For an extract query of ID queried by $\mathcal{I}$, $\mathcal{C}$ generates a random integer $a_\mathcal{I} \xleftarrow{R} \mathbb{Z}_q$ and then computes $y_{1,\mathcal{I}} = g^{a_\mathcal{I}}$ and $y_{2,\mathcal{I}} = h^{a_\mathcal{I}}$. $\mathcal{C}$ then sends $(u_{pk_\mathcal{I}}, u_{sk\mathcal{I}})$ to $\mathcal{I}$.
Identification Query. For an identification query on ID queried by $\mathcal{I}$, $\mathcal{C}$ checks if ID has been queried an extract query before. If so, $\mathcal{C}$ uses the existing $(u_{pk_\mathcal{I}}, u_{sk\mathcal{I}})$ to return a valid

transcript/conversation for $\mathcal{I}$; else, $\mathcal{C}$ runs extract query algorithm to generate (u_{pk_I}, u_{sk_I}) and returns produces a valid transcript/conversation with $\mathcal{I}$.
Hash Query. In response to query $H(PK, A_1', B_1', m_1)$, $\mathcal{C}$ checks if the hash value for m_1 is predetermined. If so, $\mathcal{C}$ returns the predetermined value to $\mathcal{I}$; else $\mathcal{C}$ generates a random value that is chosen uniformly at $\mathbb{Z}_q$.

Sign Query. In response to a signature query by $\mathcal{I}$, $\mathcal{C}$ generates random values $c'', s'' \overset{R}{\leftarrow} \mathbb{Z}_q$ and computes $A_1'' = g^{s''} {y_1}^{c''}, B_1'' = h^{s''} {y_2}^{c''}$. $\mathcal{C}$ sets $H(PK, A_1', B_1', m_1) = c''$ and outputs the signature as $\sigma'' = (c'', s'')$.

Phase 2
$\mathcal{I}$ pretends to be a valid group member using ID^*, where ID^* was queried during the extract query. After $\mathcal{I}$ generates a signature $\sigma_{\mathcal{I}} = (c_{\mathcal{I}}, s_{\mathcal{I}})$ and then sends $\sigma_{\mathcal{I}}$ to $\mathcal{C}$. As $\mathcal{C}$ obtains the signature, $\mathcal{C}$ does checking on the validity of the signature. If the signature produced by $\mathcal{I}$ is not valid, $\mathcal{C}$ aborts and it fails in the security game. Else, $\mathcal{C}$ can determine whether the given tuple is a valid DH tuple, and wins in the security game with probability as follows:

$$\begin{aligned}\Pr[\mathcal{C} \text{ wins}] &= \Pr[\text{Signature } \sigma_{\mathcal{I}} \text{ is valid}] - \Pr[\mathcal{C} \text{ aborts if it is a DH tuple}] \\ &\quad - \Pr[\mathcal{C} \text{ not aborts if it is a random tuple}] \\ &\leqslant \varepsilon_{\text{Group-IBI}} - \Pr[\mathcal{C} \text{ aborts if it is a DH tuple}] \\ &\quad - \Pr[\mathcal{C} \text{ not aborts if it is a random tuple}]\end{aligned}$$

We examine the probability that $\mathcal{C}$ aborts if the given tuple is a DH tuple. If the tuple is a valid DH tuple, $\mathcal{C}$ is able to simulate the game perfectly from the setup to sign query in Phase 1 with a negligible probability of q^{-1} on the collision of the hash oracle when answering $\mathcal{I}$s extract queries for $(q_e + 1)$ times.

Whereas in Phase 2, If the tuple is a random tuple, $\mathcal{C}$ does not abort with a probability of q^{-1} when responding to $\mathcal{I}$'s queries for q_e times. Therefore, by combining the probabilities, we get the result as follows:

$$\begin{aligned}\Pr[\mathcal{C} \text{ wins}] &\leqslant \varepsilon_{\text{Group-IBI}} - (q_e + 1)q^{-1} - (q_e + 1)q^{-1} \\ \varepsilon_{DDH} &\leqslant \varepsilon_{\text{Group-IBI}} - 2(q_e + 1)q^{-1}\end{aligned}$$

Based on the probability calculation, we are able to obtain Theorem 5.1, as required.

The time $t_{\text{Group-IBI}}$ is therefore the equivalent of running the DDH challenger simulator $\mathcal{C}$ with the addition of $q_e + 1$ total extract queries during both phases, multiplied by the two components of $(u_{pk_{\mathcal{I}}}, u_{sk_{\mathcal{I}}})$ in the extract query that require exponentiation at time $1.2t_{exp}$, thereby giving us $t_{\text{Group-IBI}} \geqslant t_{\text{DDH}} + 2.4(q_e + 1)t_{exp}$. □

5.2.2 Security Against Impersonation as $\mathcal{GM}$

Based on the proof defined in Section 4.4.2, the security proof for impersonation as a $\mathcal{GM}$ for the Group-IBI instantiated with Tan et al.'s Schnorr IBI and Katz-Wang's Schnorr DS follows vis-à-vis Tan et al.'s security proof under active and concurrent attacks. We omit this section due to lack of space, but reserve its presentation in the full version of the paper.

6 CONCLUSION

In this paper, we have presented the Group-IBI scheme, where the scheme is viewed as a combination of IBI and DS. The proposed Group-IBI enables easy members' registration and revocation within a group with the role of a group manager. Members' interactions are also not affected by the groups external interactions. Instead, the group manager is responsible for performing a verification with a verifier as a whole entity, thus providing a proof of consensus. To our knowledge, there are no existing consensus protocols yet that provide authentication using zero-knowledge proofs of knowledge.

In recent work, Ng et al. (2017) showed that by using different hard problems such as the Decisional Square Diffie-Hellman (D-Square-DH) problem, the number of public keys used in the Schnorr DS can be reduced by one. Thus, by reducing the length of the overall public keys used in the Schnorr DS. It is stated that the number of public keys can be reduced using the same method in Tan et al. (2011) 's work. Therefore, the number of public keys used in the Group-IBI can be reduced by applying a different hard problem.

We are unable to show any efficiency analysis comparisons due to this work being the first of its kind. Future work would be to construct a pairing-based instantiation using short signatures of the Boneh-Lynn-Shacham signature with tight security (Boneh et al., 2004) combined with the recent tight-IBI based on Kurosawa-Heng's original IBI by (Chia and Chin, 2020).

ACKNOWLEDGMENTS

The authors would like to thank the Multimedia University for the financial support through the Graduate Research Assistant scheme for the first author. The authors would also like to thank the Ministry of Education of Malaysia in providing financial support for this work through the Fundamental Research Grant Scheme (FRGS/1/2019/ICT04/MMU/02/5).

REFERENCES

Barapatre, P. and Rangan, C. P. (2013). Identity-based identification schemes from id-kems. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 111–129. Springer.

Bellare, M., Micciancio, D., and Warinschi, B. (2003). Foundations of group signatures: formal definition, simplified requirements and a construction based on trapdoor permutations. In *Advances in cryptology - EUROCRYPT 2003*, pages 614–629. Springer.

Bellare, M., Namprempre, C., and Neven, G. (2009). Security proofs for identity-based identification and signature schemes. *Journal of Cryptology*, 22(1):1–61.

Boneh, D. (1998). The decision diffie-hellman problem. In *International Algorithmic Number Theory Symposium*, pages 48–63. Springer.

Boneh, D. and Franklin, M. (2003). Identity-based encryption from the weil pairing. *SIAM journal on computing*, 32(3):586–615.

Boneh, D., Lynn, B., and Shacham, H. (2004). Short signatures from the weil pairing. *Journal of cryptology*, 17(4):297–319.

Camenisch, J. and Michels, M. (1998). A group signature scheme with improved efficiency. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 160–174. Springer.

Camenisch, J. and Michels, M. (1999). Separability and efficiency for generic group signature schemes. In *Annual International Cryptology Conference*, pages 413–430. Springer.

Camenisch, J. and Stadler, M. (1997). Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, pages 410–424. Springer.

Chaum, D. and Pedersen, T. P. (1992). Transferred cash grows in size. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 390–407. Springer.

Chen, L. and Pedersen, T. P. (1994). New group signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 171–181. Springer.

Chia, J. and Chin, J. (2020). An identity based-identification scheme with tight security against active and concurrent adversaries. *IEEE Access*, 8:61711–61725.

Fiat, A. and Shamir, A. (1986). How to prove yourself: Practical solutions to identification and signature problems. In *Advances in CryptologyCRYPTO86*, pages 186–194. Springer.

Gordon, S. D., Katz, J., and Vaikuntanathan, V. (2010). A group signature scheme from lattice assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–412. Springer.

Katz, J. and Wang, N. (2003). Efficiency improvements for signature schemes with tight security reductions. In *Proceedings of the 10th ACM conference on Computer and communications security* (pp. 155–164). ACM. ACM.

Kiayias, A. and Yung, M. (2005). Group signatures with efficient concurrent join. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 198–214. Springer.

Kurosawa, K. and Heng, S.-H. (2004). From digital signature to ID-based identification/signature. In *International Workshop on Public Key Cryptography*, pages 248–261.

Langlois, A., Ling, S., Nguyen, K., and Wang, H. (2014). Lattice-based group signature scheme with verifier-local revocation. In *International Workshop on Public Key Cryptography*, pages 345–361. Springer.

Lysyanskaya, A. and Ramzan, Z. (1998). Group blind digital signatures: A scalable solution to electronic cash. In *International Conference on Financial Cryptography*, pages 184–197. Springer.

Ng, T. S., Tan, S. Y., and Chin, J. J. (2017). A Variant of Schnorr Signature Scheme With Tight Security Reduction. In *Proceedings of the 8th International Conference on ICT Convergence (ICTC2017)* (pp. 411–415). IEEE. IEEE.

Schnorr, C.-P. (1989). Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer.

Tan, S.-Y., Heng, S.-H., Phan, R. C.-W., and Goi, B.-M. (2011). A variant of schnorr identity-based identification scheme with tight reduction. In *International Conference on Future Generation Information Technology*, pages 361–370. Springer.

Vangujar, A., Chin, J., Tan, S., and Ng, T. (2019). A hierarchical identity-based identification scheme without pairing. *Malaysian Journal of Mathematical Sciences*, 13(S):93–109.

A New Chaotic Map Based on Logistic and Beta Maps

Kuan-Wai Wong[*1], **Wun-She Yap**[1], **Bok-Min Goi**[1], and **Denis C.-K. Wong**[1]

[1]*Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Jalan Sungai Long, Bandar Sungai Long, 43000 Kajang, Selangor, Malaysia*

E-mail: wongkw@utar.edu.my
[*]*Corresponding author*

ABSTRACT

This paper presents a new chaotic map, called Logistic-Beta map, which is based on classical Logistic map and Beta map. Performance analysis demonstrates that it possesses a wider chaotic range, larger Lyapunov exponent and more complex chaotic behavior as compared to recent proposed chaotic map. These properties allow the proposed chaotic map suitable to use in designing image encryption scheme.

Keywords: Chaos, Logistic map, Beta Chaotic map

1 INTRODUCTION

Over the past decade, chaotic systems have received attentions from many researchers to study their chaotic behaviors. This is due to the interesting characteristics of chaotic systems, for example, aperiodicity, high sensitivity to the initial conditions and system parameters, ergodicity and random-like behaviors. This is just analogous to the confusion and diffusion properties of cryptographic properties (Shannon, 1949). Matthews (1989) was the first person to apply chaotic system to image encryption technology. Since then, the popularity of using chaos in cryptography has been grew significantly.

Chaotic system has been widely applied in designing image encryption scheme. This is because the conventional encryption methods such as Data Encryption Standard (DES) (National Bureau of Standards, 1977), Advanced Encryption Standard (AES) (Daemen and Rijmen, 2013), and International Data Encryption Algorithm (IDEA) (Lai and Massey, 1990) are no longer suitable to encrypt image data because of the bulky data capacity and high correlation among the pixels. Chaotic map is therefore applied in (a) constructing permutation matrices in the encryption process; (b) generating a chaotic pseudorandom sequences; and (c) producing the ciphertext by having the plain pixel to be the secret keys and the chaotic map to be the encryption operation (Zhang et al., 2012).

Wu et al. (2018) designed an image encryption based on a chaotic map which is formed by combining 2D-Henon map and a Sine map. The authors used the chaotic map to generate keystream and then apply DNA approach to encrypt the plain image. An image encryption scheme designed based on 2D Logistic-Sine-Cosine map was presented by Huang (2019). The chaotic system are created based on 2D Logistic, Sine and Cosine maps. Zhu et al. (2019) presented a new chaotic map based on 2D Logistic-Modulated-Sine-Coupling-Logistic map for image encryption, whereby Sine map is modulated by the Logistic map and then the result of modulation and Sine map are coupled together.

In this paper, we introduce a new chaotic map, called Logistic-Beta map which is formed by combining Logistic map with Beta map. Logistic map is a one-dimensional map which has been widely used in encryption scheme (May, 1976). Beta map is a chaotic map proposed by Zahmoul et al. (2017), which is based on a statistical distribution, called Beta function. We study the chaotic behaviors of the Logistic-Beta map, i.e. its trajectory, bifurcation diagram and Lyapunov exponent. We also demonstrate our proposed chaotic map has a better chaotic behaviors than the classical Logistic and Beta maps, and a one-dimensional logistic-based chaotic map.

2 PRELIMINARIES

This section briefly discusses the Logistic map and Beta map which are going to generate our proposed chaotic map. We also discuss a one-dimensional chaotic map that designed based on Logistic map. We will compare the chaotic behaviors of our proposed chaotic map with the following three chaotic maps in next section.

2.1 Logistic map

Logistic map is a one-dimensional discrete-time dynamical system proposed by May (1976). It is an iterated map that represented by a first order difference equations as follows.

$$x_{n+1} = rx_n(1 - x_n), \tag{1}$$

where $x_n \in (0, 1)$ and $r \in [0, 4]$.

2.2 Beta map

Zahmoul et al. (2017) proposed a chaotic map based on a Beta function, known as Beta map. It is defined as follows.

$$y_{n+1} = \mu \cdot B(y_n; y_1, y_2, c, d), \tag{2}$$

where $B(y_n; y_1, y_2, c, d)$ denotes the Beta function for $y = \{y_n\}_{n=0}^{\infty}$ and μ is a multiplier that controls the amplitude of Beta map. The beta function of y is represented by the following equation.

$$B(y; y_1, y_2, c, d) \quad = \begin{cases} \big(\frac{y-y_1}{y_m-y_1}\big)^c \big(\frac{y_2-y}{y_2-y_m}\big)^d, & \text{if } y \in (y_1, y_2); \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Given that $y_m = \frac{cy_2+dy_1}{c+d}$ denotes the weighted mean of y_1 and y_2, where $c, d, y_1, y_2 \in \mathbb{R}$ and $y_1 < y_2$. The parameters c and d are determined as follows.

$$c \quad = p_1 + q_1 \times e; \tag{4}$$

$$d \quad = p_2 + q_2 \times e, \tag{5}$$

where e is a bifurcation parameter and p_1, p_2, q_1 and $q_2 \in \mathbb{R}$ are randomly chosen constants.

A chaotic map must be bounded. To prove this, we identify the value of μ that results in $y_{n+1} \in (y_1, y_2)$. We know that the first derivative test can help to find the minima and maxima of a function, then we compute $\frac{dy_{n+1}}{dy_n} = 0$ by fixing y_1, y_2, y_m as constant.

$$\begin{aligned} \tfrac{dy_{n+1}}{dy_n} = \quad & \mu\bigg[d\Big(\frac{y_n - y_1}{y_m - y_1}\Big)^c\Big(\frac{y_2 - y_n}{y_2 - y_m}\Big)^{d-1}\Big(-\frac{1}{y_2 - y_m}\Big) + \\ & c\Big(\frac{y_2 - y_n}{y_2 - y_m}\Big)^d\Big(\frac{y_n - y_1}{y_m - y_1}\Big)^{c-1}\Big(\frac{1}{y_m - y_1}\Big)\bigg] \\ 0 = \quad & \mu \cdot \frac{(y_n - y_1)^{c-1}(y_2 - y_n)^{d-1}}{(y_m - y_1)^c(y_2 - y_m)^d} \cdot [c(y_2 - y_n) - d(y_n - y_1)] \\ y_n = \quad & \frac{cy_2 + dy_1}{c + d}. \end{aligned} \tag{6}$$

Noted that $y_n = y_m$. Next, we compute the second derivative on y_{n+1} with respect to y_n as follows.

$$\begin{aligned} \frac{d^2y_{n+1}}{dy_n^2} = \quad & \mu\big(\tfrac{y_n-y_1}{y_m-y_1}\big)^{c-2}\big(\tfrac{y_2-y_n}{y_2-y_m}\big)^{d-2}\big(\tfrac{1}{(y_2-y_m)(y_m-y_1)}\big)^2 \cdot \\ & \big(d(y_n - y_1)[-c(y_2 - y_n) + (d-1)(y_n - y_1)] + \\ & c(y_2 - y_n)[-d(y_n - y_1) + (c-1)(y_2 - y_n)]\big) \end{aligned} \tag{7}$$

Then, substitute Eq. (6) into Eq. (7).

$$\begin{aligned} \frac{d^2y_{n+1}}{dy_n^2}\bigg|_{y_n=ym} = \quad & \mu\big(\tfrac{1}{(y_2-y_m)(y_m-y_1)}\big)^2 \cdot \big[cd(y_m - y_1)(y_1 - y_2) \\ & +cd(y_2 - y_m)(y_1 - y2)\big] < 0, \quad \because y_1 < y_2. \end{aligned} \tag{8}$$

Therefore, y_m is the local maximum.

By letting $y_n = y_m$, we determine the range of μ by substituting Eq. (6) into Eq. (2) and (3) as follows.

$$y_1 < \mu \cdot \big(\tfrac{y_m-y_1}{y_m-y_1}\big)^c\big(\tfrac{y_2-y_m}{y_2-y_m}\big)^d < y_2. \tag{9}$$

Hence, $\mu \in (y_1, y_2)$.

2.3 Modified Logistic map

The is a chaotic map designed by modifying the Logistic map discussed in Subsection 2.1. It is proposed by Lestari et al. (2018) to allow the initial values to be positive or negative. It can be defined as follows.

$$x_{n+1} = \begin{cases} g_1(x_n); \\ h_1(x_n), \end{cases} \tag{10}$$

for $x_n \in (-1, 1)$.

The recursive equation of the modification is given as follows.

$$x_{n+1} = \begin{cases} (-\frac{3}{2}|r| - \sqrt{2}|r|) \cdot x_n \cdot ((2\sqrt{2}-2)x_n + 1), & \text{for } -1 < x_n < 0; \\ (-\frac{3}{2}|r| - \sqrt{2}|r|) \cdot x_n \cdot ((2\sqrt{2}-2)x_n - 1), & \text{for } 0 \leq x_n < 1, \end{cases} \tag{11}$$

where $r \in [-4, 4]$. In Section 4, we compare the dynamical performance of this chaotic map with our proposed map.

3 THE PROPOSED CHAOTIC MAP

The newly proposed chaotic map, called Logistic-Beta map is designed by combining of two chaotic maps, i.e. Logistic map and Beta map. The mathematical model of our new one-dimensional chaotic map is based on the following equation.

$$x_{n+1} = f(x_n) = g(h(x_n)), \text{and } f : [0,1] \rightarrow [0,1].$$

where $h(\cdot)$ represents the Beta map with $\mu = 1$ given in Eq. (2). Beta map is chosen to enlarge phase space, while $g(\cdot)$ represents the Logistic map given in Eq. (1). Therefore, the Logistic-Beta map is defined as follows.

$$x_{n+1} = r\left(\frac{x_n - y_1}{y_m - y_1}\right)^c \left(\frac{y_2 - x_n}{y_2 - y_m}\right)^d \left[1 - \left(\frac{x_n - y_1}{y_m - y_1}\right)^c \left(\frac{y_2 - x_n}{y_2 - y_m}\right)^d\right], \tag{12}$$

where n is the iteration number, $y_m = \frac{cy_2 + dy_1}{c+d}$ and $c, d, y_1, y_2 \in \mathbb{R}$ and $y_1 < y_2$. Recall that parameters c and d are determined by equations (4) and (5) as follows.

$$\begin{aligned} c &= p_1 + q_1 \times e; \\ d &= p_2 + q_2 \times e, \end{aligned}$$

where e is a bifurcation parameter and p_1, p_2, q_1 and q_2 are randomly chosen constants.

Since the chaotic map must be bounded, careful selection of the parameter r must be done to ensure the phase space is in a closed interval. Rewrite Eq. (12) as follows.

$$x_{n+1} = rh(x_n)(1 - h(x_n)) = rh(x_n) - r[h(x_n)]^2, \tag{13}$$

where $h(x_n) = \left(\frac{x_n - y_1}{y_m - y_1}\right)^c \left(\frac{y_2 - x_n}{y_2 - y_m}\right)^d$. To obtain the maximum value of x_{n+1}, solve x_n in the following equation.

$$\frac{dx_{n+1}}{dx_n} = rh'(x_n) - 2rh(x_n) \cdot h'(x_n) = rh'(x_n)[1 - 2h(x_n)] = 0, \tag{14}$$

where

$$\begin{aligned} h'(x_n) = & \quad d\big(\tfrac{x_n-y_1}{y_m-y_1}\big)^c\big(\tfrac{y_2-x_n}{y_2-y_m}\big)^{d-1}\big(-\tfrac{1}{y_2-y_m}\big)+ \\ & \quad c\big(\tfrac{y_2-x_n}{y_2-y_m}\big)^d\big(\tfrac{x_n-y_1}{y_m-y_1}\big)^{c-1}\big(\tfrac{1}{y_m-y_1}\big) \\ = & \quad \tfrac{(x_n-y_1)^{c-1}(y_2-x_n)^{d-1}}{(y_m-y_1)^c(y_2-y_m)^d}\cdot[c(y_2-x_n)-d(x_n-y_1)]. \end{aligned} \tag{15}$$

When $r=0$, $x_{n+1}=0$ regardless the value of x_n.

Next, when $h'(x_n)=0$, we obtain

$$x_n = \frac{cy_2+dy_1}{c+d}. \tag{16}$$

This obtained x_n equals to y_m in Eq. (12). Then, we substitute y_m into Eq. (12) and obtain

$$r\left(\frac{y_m-y_1}{y_m-y_1}\right)^c\left(\frac{y_2-y_m}{y_2-y_m}\right)^d\left[1-\left(\frac{y_m-y_1}{y_m-y_1}\right)^c\left(\frac{y_2-y_m}{y_2-y_m}\right)^d\right]=0. \tag{17}$$

Therefore, when $x_n=y_m$, we will get $x_{n+1}=0$ regardless the value of r. We could not determine the range of r for this case.

So, we look at the final case, i.e. when $1-2h(x_n)=0$. We have

$$\begin{aligned} \big(\tfrac{x_n-y_1}{y_m-y_1}\big)^c\big(\tfrac{y_2-x_n}{y_2-y_m}\big)^d &= \frac{1}{2} \\ (x_n-y_1)^c(y_2-x_n)^d &= \frac{1}{2}(y_m-y_1)^c(y_2-y_m)^d \\ &= \big(\frac{1}{2^{1/2c}}(y_m-y_1)\big)^c\big(\frac{1}{2^{1/2d}}(y_2-y_m)\big)^d \\ &= \left[\left(\frac{1}{2^{1/2c}}y_m+\big(1-\frac{1}{2^{1/2c}}\big)y_1\right)-y_1\right]^c\times \\ & \quad \left[y_2-\left(\big(1-\frac{1}{2^{1/2d}}\big)y_2+\frac{1}{2^{1/2d}}y_m\right)\right]^d. \end{aligned} \tag{18}$$

So,

$$x_n=\frac{1}{2^{1/2c}}y_m+\big(1-\frac{1}{2^{1/2c}}\big)y_1=\big(1-\frac{1}{2^{1/2d}}\big)y_2+\frac{1}{2^{1/2d}}y_m. \tag{19}$$

To make sure $x_{n+1}\in[0,1]$, we determine r by substituting Eq. (19) into Eq. (12). Let $x_{n,1}=\frac{1}{2^{1/2c}}y_m+\big(1-\frac{1}{2^{1/2c}}\big)y_1$ and $x_{n,2}=\big(1-\frac{1}{2^{1/2d}}\big)y_2+\frac{1}{2^{1/2d}}y_m$. Then, we obtain the range of r as follows.

$$\begin{aligned} 0\le r\left(\frac{x_{n,1}-y_1}{y_m-y_1}\right)^c\left(\frac{y_2-x_{n,2}}{y_2-y_m}\right)^d\left[1-\left(\frac{x_{n,1}-y_1}{y_m-y_1}\right)^c\left(\frac{y_2-x_{n,2}}{y_2-y_m}\right)^d\right]\le 1 \\ 0\le r\left(\frac{\frac{1}{2^{1/2c}}(y_m-y_1)}{y_m-y_1}\right)^c\left(\frac{\frac{1}{2^{1/2d}}(y_2-y_m)}{y_2-y_m}\right)^d\left[1-\left(\frac{\frac{1}{2^{1/2c}}(y_m-y_1)}{y_m-y_1}\right)^c\left(\frac{\frac{1}{2^{1/2d}}(y_2-y_m)}{y_2-y_m}\right)^d\right]\le 1 \\ 0\le r\big(\frac{1}{\sqrt{2}}\big)\big(\frac{1}{\sqrt{2}}\big)\big(1-\big(\frac{1}{\sqrt{2}}\big)\big(\frac{1}{\sqrt{2}}\big)\big)\le 1 \\ 0\le r\le 4. \end{aligned} \tag{20}$$

From Eq. (20), we have $r\in[0,4]$ which is same as the Logistic map.

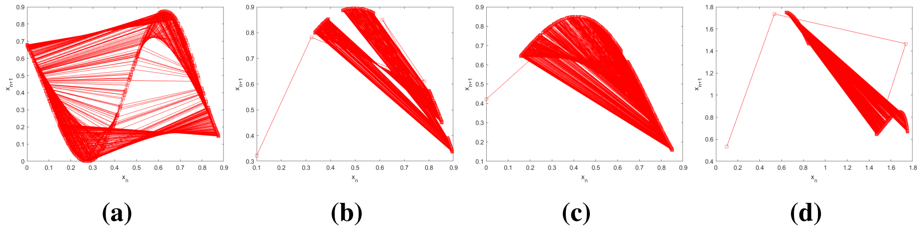
4 DYNAMICAL PERFORMANCE

In this section, we will characterize the dynamics of Logistic-Beta map geometrically with the trajectory and bifurcation plots, and statistically with the Lyapunov exponent.

4.1 Trajectory

Trajectory or orbit presents the moving path of the set of all points in the dynamical system (Kocarev and Lian, 2011). We show the trajectories for the Logistic-Beta map and the chaotic maps discussed in Section 2.

For Logistic-Beta and Beta maps, we set the initial values, $x_0 = 0$, as shown in Figures 1a and 1c. While the initial values for Logistic map and modified Logistic maps, $x_0 = 0.1$ and their trajectories are plotted in Figures 1b and 1d. As shown in Figure 1a, Logistic-Beta map has a larger distribution area as compared to Logistic and Beta maps, referring to Figures 1b and 1c. Even though the modified Logistic map in Figure 1d has a wider range for x_{n+1}, the outputs are not random and lack of dispersion. Therefore, Logistic-Beta map produces more random output and demonstrates a better ergodicity.

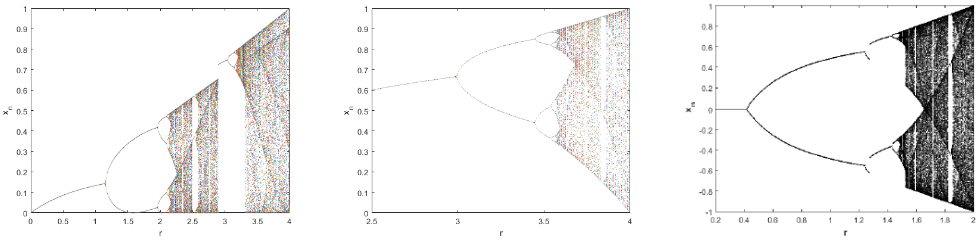


(a) (b) (c) (d)

Figure 1: Trajectory Diagram: **(a)** Logistic-Beta map with $r = 3.5, e = 0.1, y_1 = -1, y_2 = 1, p_1 = 5, p_2 = 3, q_1 = 1, q_2 = -1$ **(b)** Logistic map with $r = 3.58$ **(c)** Beta-map with $\mu = 0.85, e = 0.65, y_1 = -1, y_2 = 1, p_1 = 5, p_2 = 3, q_1 = 1, q_2 = -1$ **(d)** Modified Logistic map with $r = 2$

4.2 Bifurcation diagram

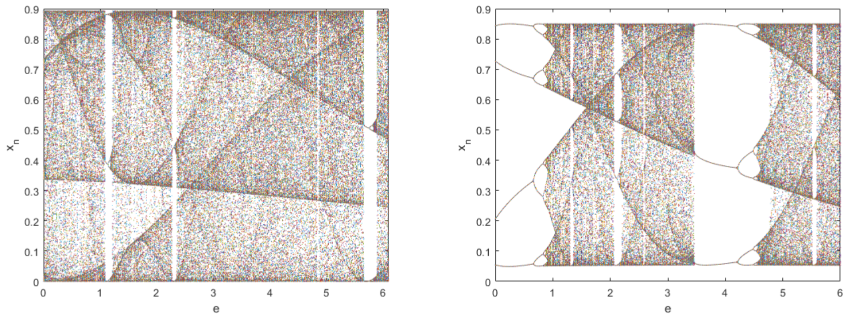
Bifurcation shows a qualitative change in dynamics for the variation of the control parameters of a dynamical system (Kocarev and Lian, 2011). In other word, the dotted area of the diagram describes the chaotic behavior of the system. As shown in Eq. (12), Logistic-Beta map consists of two control parameters, i.e. r from Logistic map in Eq. (1) and e from Beta map in Eq. (4) and (5). We first vary the parameter r and shows the bifurcation diagram for $r = [0, 4]$ in Figure 2a. When the parameters exceed the critical value, i.e. $r = 1.155$, the Logistic-Beta map exhibits a period-doubling bifurcation by converting the attractor from a period-1 firing to a period-2 firing. The following period-doubling bifurcations occur at $r = 1.95, 2.1, 2.53$ and 3.04. The dotted area in between bifurcations shows that the onset of chaos as various curves start merging together. As shown in Figure 2, the logistic-based chaotic maps consists of windows of periodic behaviors causing the maps vulnerable to parameter estimation attacks (Arroyo et al., 2010).



(a) (b) (c)

Figure 2: Bifurcation diagram of chaotic maps with different bifurcation parameters: **(a)** Logistics-Beta map with $0 \leq r \leq 4, e = 0.4, y_1 = -1, y_2 = 1, p_1 = 4, p_2 = 2, q_1 = 1, q_2 = 0.2$ **(b)** Logistics map with $2.5 \leq r \leq 4$ **(c)** Modified Logistic map with $0.2 \leq r \leq 2$

Logistic-Beta map has an advantage over the other maps as it has another control parameter e which enlarges the phase space and make the proposed map more chaotic. We compare the bifurcation diagram of Beta map and Logistic-Beta map by varying parameters e, refer to Figures 3 for the comparison. As shown in Figure 3a, the proposed map has excellent chaotic behavior along the range $e \in [0, 6]$ as it has a very few periodic windows as compared to Beta and the dotted points are scattered around the area.

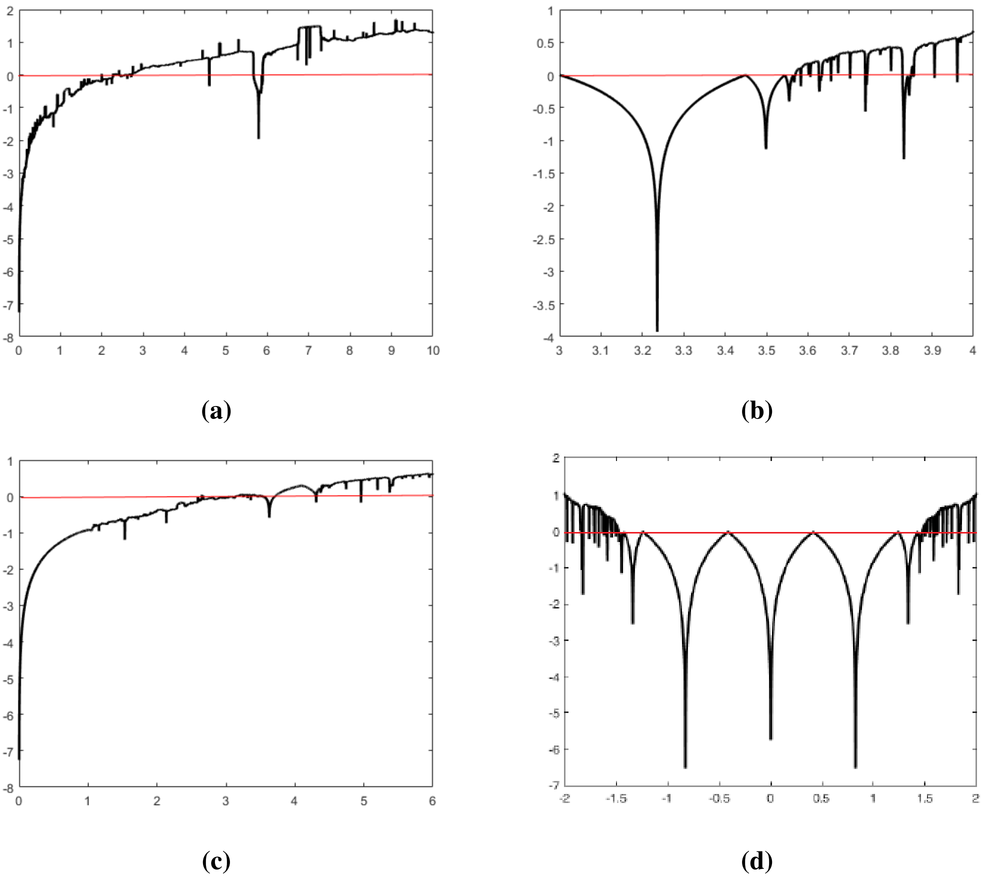


(a) (b)

Figure 3: Bifurcation diagram of chaotic maps with different bifurcation parameters: **(a)** Logistics-Beta map with $0 \leq e \leq 9$ and $r = 3.57, y_1 = -1, y_2 = 1, p_1 = 4, p_2 = 2, q_1 = 1, q_2 = 0.2$ **(b)** Beta map with $\mu = 0.85, y_1 = -1, y_2 = 1, p_1 = 4, p_2 = 2, q_1 = 1, q_2 = 0.2$

4.3 Lyapunov Exponent

Lyapunov Exponent (LE) is a quantitative measure to test the sensitivity of the chaotic map to the slight changes in the initial conditions and control parameters (Kocarev and Lian, 2011). A positive LE indicates that the chaotic map has a good chaotic behavior, and the higher the LE value shows a better sensitivity of the map to its initial value or system parameters. From Figure 4a, it is obvious that Logistic-Beta map has the highest LE value and also a greater chaotic range, i.e. it has positive LE for $e > 2.3$. As shown in Figures 4b and 4c, Logistic map has positive LE for $r \in [3.57, 4]$ while Beta map has positive LE values for $e > 3.2$. For modified Logistic map in Figure 4d, the chaotic map only have positive LE when $r \in [-2, -1.5] \cup [1.5, 2]$.



(a) **(b)**

(c) **(d)**

Figure 4: Lyapunov Exponent: **(a)** Logistic-Beta map with $e \in [0, 10], r = 3.57, y_1 = -1, y_2 = 1, p_1 = 4, p_2 = 2, q_1 = 1, q_2 = 0.2$, **(b)** Logistic map with $r \in [3, 4]$, **(c)** Beta-map with $e \in [0, 6], \mu = 0.85, y_1 = -1, y_2 = 1, p_1 = 4, p_2 = 2, q_1 = 1, q_2 = 0.2$, **(d)** Modified Logistic map with $r = \pm 2$

5 CONCLUSION

This paper proposes a new chaotic map, called Logistic-Beta map. We have proven that the proposed chaotic map has significantly improved the chaotic behaviors of classical Logistic and Beta maps. The chaotic behaviors of proposed chaotic map also have been discussed and compared with modified Logistic map which is a chaotic map designed based on Logistic map. The advantages of Logistic-Beta map is summarized as follows.

1. The large distribution area in the phase plane shows that our map has a good ergodicity.
2. The large darked area in the bifurcation diagram demonstrates that the proposed map has a large chaotic region, leading to a large key space.

3. A positive Lyapunov Exponent value indicates that our proposed map has good sensitivity to initial values.

These advantages make the proposed chaotic map suitable to be applied in an image encryption scheme.

REFERENCES

Arroyo, D., Amigó Garcia, J. M., Li, S., and Alvarez, G. (2010). *On the inadequacy of unimodal maps for cryptographic applications*.

Daemen, J. and Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

Huang, H. (2019). Novel scheme for image encryption combining 2d logistic-sine-cosine map and double random-phase encoding. *IEEE Access*, 7:177988–177996.

Kocarev, L. and Lian, S. (2011). *Chaos-based cryptography: Theory, algorithms and applications*, volume 354. Springer Science & Business Media.

Lai, X. and Massey, J. L. (1990). A proposal for a new block encryption standard. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 389–404. Springer.

Lestari, A. A., Suryadi, M., Ramli, K., et al. (2018). Modified logistic maps for discrete time chaos based random number generator. In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 391–396. IEEE.

Matthews, R. (1989). On the derivation of a "chaotic" encryption algorithm. *Cryptologia*, 13(1):29–42.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467.

National Bureau of Standards (1977). *Data Encryption Standard*. U.S. Department of Commerce, FIPS-Pub.46.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715.

Wu, J., Liao, X., and Yang, B. (2018). Image encryption using 2d hénon-sine map and dna approach. *Signal Processing*, 153:11–23.

Zahmoul, R., Ejbali, R., and Zaied, M. (2017). Image encryption based on new beta chaotic maps. *Optics and Lasers in Engineering*, 96:39–49.

Zhang, L. Y., Li, C., Wong, K.-W., Shu, S., and Chen, G. (2012). Cryptanalyzing a chaos-based image encryption algorithm using alternate structure. *Journal of Systems and Software*, 85(9):2077–2085.

Zhu, H., Zhao, Y., and Song, Y. (2019). 2d logistic-modulated-sine-coupling-logistic chaotic map for image encryption. *IEEE Access*, 7:14081–14098.

No-Go Theorems for Data Privacy

Thomas Studer[*1]

[1]*Institute of Computer Science, University of Bern, Switzerland*

E-mail: thomas.studer@inf.unibe.ch
[*]*Corresponding author*

ABSTRACT

Controlled query evaluation (CQE) is an approach to guarantee data privacy for database and knowledge base systems. CQE-systems feature a censor function that may distort the answer to a query in order to hide sensitive information. We introduce a high-level formalization of controlled query evaluation and define several desirable properties of CQE-systems. Finally we establish two no-go theorems, which show that certain combinations of these properties cannot be obtained.

Keywords: Impossibility theorem, data privacy, controlled query evaluation, modal logic

1 INTRODUCTION

Controlled query evaluation (CQE) refers to a data privacy mechanism where the database (or knowledge base) is equipped with a censor function. This censor checks for each query whether the answer to the query would reveal sensitive information to a user. If this is the case, then the censor will distort the answer. Essentially, there are two possibilities how an answer may be distorted:

1. the CQE-system may refuse to answer the query (Sicherman et al., 1983) or
2. the CQE-system may give an incorrect answer, i.e. it lies (Bonatti et al., 1995).

This censor based approach has the advantage that the task of maintaining privacy is separated from the task of keeping the data. This gives more flexibility than an integrated approach (like hiding rows in a database) and guarantees than no information is leaked through otherwise unidentified inference channels. Controlled query evaluation has been applied to a variety of data models and control mechansims, see, e.g. Refs. Biskup (2000), Biskup and Bonatti (2001, 2004a,b), Biskup and Weibert (2008), Studer and Werner (2014).

No-go theorems are well-known in theoretical physics where they describe particular situations that are not physically possible. Often the term is used for results in quantum mechanics like Bell's theorem (Bell, 1964), the Kochen–Specker theorem (Kochen and Specker, 1968), or, for a more recent example, the Frauchiger–Renner paradox (Frauchiger and Renner, 2018). Nurgalieva and del Rio (2019) provide a modal logic analysis of the latter paradox. Arrow's theorem (Arrow, 1950) in social choice theory also is a no-go theorem stating that no voting system can be designed that meets certain given fairness conditions. Pacuit and Yang (2016) present a version of independence logic in which Arrow's theorem is derivable.

In the present paper we develop a highly abstract model for dynamic query evaluation systems like CQE. We formulate several desirable properties of CQE-systems in our framework and establish two no-go theorems saying that certain combinations of those properties are impossible. The main contribution of this paper is the presentation of the abstract logical framework as well as the high-level formulation of the no-go theorems. Note that some particular instances of our results have already been known (Biskup, 2000, Studer and Werner, 2014).

There are many different notions of privacy available in the literature. For our results, we rely on *provable privacy* (Stoffel and Studer, 2005, Stouppa and Studer, 2007), which is a rather weak notion of data privacy. Note that using a weak definition of privacy makes our impossibility theorems actually stronger since they state that under certain conditions not even this weak form of privacy can be achieved.

Clearly our work is also connected to the issues of lying and deception. Logics dealing with these notions are introduced and studied, e.g., by Ågotnes et al. (2018), Icard (2019), van Ditmarsch (2014).

In this version of the paper, we had to omit all proofs for lack of space. A version with full proofs is available in Ref. Studer (2020).

2 LOGICAL PRELIMINARIES

Let X be a set. We use $\mathcal{P}(X)$ to denote the power set of X. For sets Γ and Δ we use Γ, Δ for $\Gamma \cup \Delta$. Moreover, in such a context we write A for the singleton set $\{A\}$. Hence Γ, A stands for $\Gamma \cup \{A\}$.

Definition 2.1. *A* logic L *is given by*

1. *a set of formulas* Fml_L *and*
2. *a consequence relation* $\vdash_\mathsf{L}$ *for* L *that is a relation between sets of formulas and formulas, i.e.* $\vdash_\mathsf{L} \subseteq \mathcal{P}(\mathsf{Fml}_\mathsf{L}) \times \mathsf{Fml}_\mathsf{L}$ *satisfying for all* $A, C \in \mathsf{Fml}_\mathsf{L}$ *and* $\Gamma, \Delta \in \mathcal{P}(\mathsf{Fml}_\mathsf{L})$*:*
   (a) *reflexivity:* $\{A\} \vdash_\mathsf{L} A$*;*
   (b) *weakening:* $\Gamma \vdash_\mathsf{L} A \implies \Gamma, \Delta \vdash_\mathsf{L} A$*;*
   (c) *transitivity:* $\Gamma \vdash_\mathsf{L} C$ *and* $\Delta, C \vdash_\mathsf{L} A \implies \Gamma, \Delta \vdash_\mathsf{L} A$.

Transitivity is sometimes called *cut*. The previous definition gives us single conclusion consequence relations, which is sufficient for the purpose of this paper. For other notions of consequence relations see, e.g., Refs. Avron (1991) and Iemhoff (2016).

As usual, we write $\vdash_{\mathsf{L}} A$ for $\emptyset \vdash_{\mathsf{L}} A$. A formula A is called a *theorem of* L if $\vdash_{\mathsf{L}} A$.

We do not specify the logic L any further. The only thing we need is a consequence relation as given above. For instance, L may be classical propositional logic with $\vdash_{\mathsf{L}}$ being the usual derivation relation (see Section 4) or L may be a description logic with $\vdash_{\mathsf{L}}$ being its semantic consequence relation (Studer and Werner, 2014).

Definition 2.2.

1. *A logic* L *is called* consistent *if there exists a formula* $A \in \mathsf{Fml}_{\mathsf{L}}$ *such that* $\nvdash_{\mathsf{L}} A$.
2. *A set* Γ *of* $\mathsf{Fml}_{\mathsf{L}}$*-formulas is called* L-consistent *if there exists a formula* $A \in \mathsf{Fml}_{\mathsf{L}}$ *such that* $\Gamma \nvdash_{\mathsf{L}} A$.

We need a simple modal logic M over L.

Definition 2.3. *The set of formulas* $\mathsf{Fml}_{\mathsf{M}}$ *is given inductively by:*

1. *if* A *is a formula of* $\mathsf{Fml}_{\mathsf{L}}$, *then* $\Box A$ *is a formula of* $\mathsf{Fml}_{\mathsf{M}}$*;*
2. $\bot$ *is a formula of* $\mathsf{Fml}_{\mathsf{M}}$*;*
3. *if* A *and* B *are formulas of* $\mathsf{Fml}_{\mathsf{M}}$, *so is* $A \to B$, *too.*

As usual, the symbol $\bot$ denotes falsum and $\Box A$ means that A is known. We define the remaining classical connectives $\top$, $\wedge$, $\vee$, and $\neg$ in the standard way. Note that M is not a fully-fledged modal logic. For instance, it does not include nested modalities.

We give semantics to $\mathsf{Fml}_{\mathsf{M}}$-formulas as follows.

Definition 2.4. *An* M*-model* $\mathcal{M}$ *is a set of sets of* $\mathsf{Fml}_{\mathsf{L}}$*-formulas, that is*

$$\mathcal{M} \subseteq \mathcal{P}(\mathsf{Fml}_{\mathsf{L}}).$$

Definition 2.5. *Let* $\mathcal{M}$ *be an* M*-model. Truth of an* $\mathsf{Fml}_{\mathsf{M}}$*-formula in* $\mathcal{M}$ *is inductively defined by:*

1. $\mathcal{M} \Vdash \Box A$ *iff* $w \vdash_{\mathsf{L}} A$ *for all* $w \in \mathcal{M}$*;*
2. $\mathcal{M} \nVdash \bot$*;*
3. $\mathcal{M} \Vdash A \to B$ *iff* $\mathcal{M} \nVdash A$ *or* $\mathcal{M} \Vdash B$.

We use the following standard definition.

Definition 2.6. *Let* Γ *be a set of* $\mathsf{Fml}_{\mathsf{M}}$*-formulas.*

1. *We write* $\mathcal{M} \Vdash \Gamma$ *iff* $\mathcal{M} \Vdash A$ *for each* $A \in \Gamma$.
2. Γ *is called* satisfiable *iff there exists an* M*-model* $\mathcal{M}$ *with* $\mathcal{M} \Vdash \Gamma$.
3. Γ entails *a formula* A*, in symbols* $\Gamma \models A$*, iff for each model* $\mathcal{M}$ *we have that*

$$\mathcal{M} \Vdash \Gamma \quad \textit{implies} \quad \mathcal{M} \Vdash A.$$

3 PRIVACY

Definition 3.1. *A* privacy configuration *is a triple* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *that consists of:*

1. *the knowledge base* $\mathsf{KB} \subseteq \mathsf{Fml}_{\mathsf{L}}$*, which is only accessible via the censor;*
2. *the set of a priori knowledge* $\mathsf{AK} \subseteq \mathsf{Fml}_{\mathsf{M}}$*, which formalizes general background knowledge known to the attacker and the censor;*
3. *the set of secrets* $\mathsf{Sec} \subseteq \mathsf{Fml}_{\mathsf{L}}$*, which should be protected by the censor.*

A privacy configuration $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *satisfies the following conditions:*

1. KB *is* L*-consistent (consistency);*
2. $\{\mathsf{KB}\} \Vdash \mathsf{AK}$ *(truthful start);*
3. $\mathsf{AK} \not\models \Box s$ *for each* $s \in \mathsf{Sec}$ *(hidden secrets).*

Note that in the above definition, KB and Sec are sets of $\mathsf{Fml}_{\mathsf{L}}$-formulas while AK is a set of $\mathsf{Fml}_{\mathsf{M}}$-formulas. Thus AK may not only contain domain knowledge but also knowledge about the structure of KB. This is further explained in Section 4.

A *query* to a knowledge base KB is simply a formula of $\mathsf{Fml}_{\mathsf{L}}$.

Given a logic L, we can evaluate a query q over a knowledge base KB. There are two possible answers: t (true) and u (unknown).

Definition 3.2. *The evaluation function* eval *is defined by:*

$$\mathsf{eval}(\mathsf{KB}, q) := \begin{cases} t & \textit{if} \quad \mathsf{KB} \vdash_{\mathsf{L}} q \\ u & \textit{otherwise} \end{cases}$$

If the language of the logic L includes negation, then one may also consider an evaluation function that can return the value f (false), i.e. one defines $\mathsf{eval}(\mathsf{KB}, q) := f$ if $\mathsf{KB} \vdash_{\mathsf{L}} \neg q$. However, in the general setting of this paper, we cannot include this case.

A censor has to hide the secrets. In order to achieve this, it can not only answer t and u to a query but also r (refuse to answer). We denote the set of possible answers of a censor by

$$\mathbb{A} := \{t, u, r\}.$$

Let X be a set. Then X^{ω} denotes the set of infinite sequences of elements of X.

Definition 3.3. *A* censor *is a mapping that assigns an answering function*

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})} : \mathsf{Fml}_{\mathsf{L}}^{\omega} \longrightarrow \mathbb{A}^{\omega}$$

to each privacy configuration $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$. *By abuse of notation, we also call the answering function* $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$ *a* censor. *A sequence* $q \in \mathsf{Fml}_{\mathsf{L}}^{\omega}$ *is called* query sequence.

Usually, the privacy configuration will be clear from the context. In that case we simply use Cens instead of $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$.

Given a sequence s, we use s_i to denote its i-th element. That is for a query sequence $q \in \mathsf{Fml}_{\mathsf{L}}^{\omega}$, we use q_i to denote the i-th query and $\mathsf{Cens}(q)_i$ to denote the i-th answer of the censor.

Example 3.1. *Let* $A, B, C \in \mathsf{Fml}_{\mathsf{L}}$. *We define a privacy configuration with* $\mathsf{KB} = \{A, C\}$, $\mathsf{AK} = \emptyset$, *and* $\mathsf{Sec} = \{C\}$. *A censor* Cens *yields an answering function* $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$, *which applied to a query sequence* $q = (A, B, C, \ldots)$ *yields a sequence of answers, e.g.,*

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q) = t, u, r, \ldots.$$

In this case, $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$ *gives true answers since* $\mathsf{eval}(\mathsf{KB}, A) = t$ *and* $\mathsf{eval}(\mathsf{KB}, B) = u$ *and it protects the secret be refusing to answer the query* C.

Another option for the answering function would be to answer the third query with u*, i.e., it would lie (instead of refuse to answer) in order to protect the secret.*

A further option would be to always refuse the answer, i.e.

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q) = r, r, r, \ldots.$$

This, of course, would be a trivial (and useless) answering function that would, however, preserve all secrets.

In this paper, we will consider continuous censors only, which are given as follows.

Definition 3.4. *A censor* Cens *is* continuous *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and for all query sequences* $q, q' \in \mathsf{Fml}_{\mathsf{L}}^{\omega}$ *and all* $n \in \omega$ *we have that*

$$q|_n = q'|_n \quad \Longrightarrow \quad \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_n = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q')|_n$$

where for an infinite sequence $s = (s_1, s_2, \ldots)$*, we use* $s|_n$ *to denote the initial segment of* s *of length* n*, i.e.* $s|_n = (s_1, \ldots, s_n)$.

Continuity means that the answer of a censor to a query does not depend on future queries, see also Lemma 3.1.

A censor is called truthful if it does not lie.

Definition 3.5. *A censor* Cens *is called* truthful *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$*, all query sequences* $q = (q_1, q_2, \ldots)$*, and all sequences*

$$(a_1, a_2, \ldots) = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)$$

we have that for all $i \in \omega$

$$a_i = \mathsf{eval}(\mathsf{KB}, q_i) \quad or \quad a_i = r.$$

Hence a truthful censor may refuse to answer a query in order to protect a secret but it will not give an incorrect answer.

In the modal logic M over L, we can express what knowledge one can gain from the answers of a censor to a query. This is called the content of the answer.

Definition 3.6. *Given an answer* $a \in \mathbb{A}$ *to a query* $q \in \mathsf{Fml}_\mathsf{L}$*, we define its* content *as follows:*

$$\mathsf{cont}(q, t) := \Box q$$
$$\mathsf{cont}(q, u) := \neg\Box q$$
$$\mathsf{cont}(q, r) := \top$$

Assume that we are given a privacy configuration $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and a censor* Cens*. We define the content of the answers of the censor to a query sequence* $q \in \mathsf{Fml}_\mathsf{L}^\omega$ *up to* $n \in \omega$ *by*

$$\mathsf{cont}(\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n) := \bigcup_{1 \le i \le n} \{\mathsf{cont}(q_i, a_i)\} \cup \mathsf{AK}$$

where $a = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)$*. Note that here we have also included the a priori knowledge.*

The following is a trivial observation showing the role of continuity.

Lemma 3.1. *Let* Cens *be a continuous censor. The content function is monotone in the second argument: for* $m \le n$ *we have*

$$\mathsf{cont}(\mathsf{Cens}(q), m) \subseteq \mathsf{cont}(\mathsf{Cens}(q), n).$$

We call a censor credible if it does not return contradicting answers.

Definition 3.7. *A censor* Cens *is called* credible *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and for every query sequence* q *and every* $n \in \omega$*, the set* $\mathsf{cont}(\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n)$ *is satisfiable.*

Definition 3.8. *The* full content *of a knowledge base* KB *is given by*

$$\mathsf{full}(\mathsf{KB}) := \bigcup_{A \in \mathsf{Fml}_\mathsf{L}} \mathsf{cont}(A, \mathsf{eval}(\mathsf{KB}, A)).$$

Lemma 3.2. *For any knowledge base* KB*, we have that*

$$\{\mathsf{KB}\} \Vdash \mathsf{full}(\mathsf{KB}).$$

Lemma 3.3. *We let* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *be a privacy configuration. Further we let* $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$ *be a truthful censor. For every query sequence* q *and* $n \in \omega$*, we have that*

$$\mathsf{cont}(\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n) \subseteq \mathsf{full}(\mathsf{KB}) \cup \{\top\} \cup \mathsf{AK}.$$

The following corollary is a generalization of Cor. 30 in Ref. Studer and Werner (2014).

Corollary 3.1. *Every truthful censor is credible.*

There are several properties that a 'good' censor should fulfil. We call a censor effective if it protects all secrets.

Definition 3.9. *A censor* Cens *is called* effective *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and for every query sequence* $q \in \mathsf{Fml}_{\mathsf{L}}^{\omega}$ *and every* $n \in \omega$*, we have*

$$\mathsf{cont}(\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n) \not\models \Box s \quad \textit{for each } s \in \mathsf{Sec}$$

A 'good' censor should only distort an answer to a query when it is absolutely necessary, i.e. when giving the correct answer would leak a secret. We call such a censor minimally invasive.

Definition 3.10. *Let* Cens *be an effective and credible censor. This censor is called* minimally invasive *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and for each query sequence* $q \in \mathsf{Fml}_{\mathsf{L}}^{\omega}$*, we have that whenever*

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i \neq \mathsf{eval}(\mathsf{KB}, q_i),$$

replacing

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i \quad \textit{with} \quad \mathsf{eval}(\mathsf{KB}, q_i)$$

would lead to a violation of effectiveness or credibility, that is for any censor $\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$ *such that*

$$\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_{i-1} = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_{i-1}$$

and

$$\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i = \mathsf{eval}(\mathsf{KB}, q_i)$$

we have that for some n

$$\mathsf{cont}(\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n) \models \Box s \quad \textit{for some } s \in \mathsf{Sec}$$

or

$$\mathsf{cont}(\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), n) \textit{ is not satisfiable}.$$

It is a trivial observation that a truthful, effective and minimally invasive censor has to answer the same query always in the same way.

Lemma 3.4. *Let* Cens *be a truthful, effective and minimally invasive censor. Further let* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *be a privacy configuration and* q *be a query sequence with* $q_i = q_j$ *for some* i, j*. Then*

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_j.$$

Consider a truthful, effective, continuous and minimally invasive censor and a given query sequence. If the censor lies to answer some query, then giving the correct answer would immediately reveal a secret.

Lemma 3.5. *Let* Cens *be a truthful, effective, continuous and minimally invasive censor. Further let* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *be a privacy configuration and* q *be a query sequence. Let* i *be the least natural number such that*

$$\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i \neq \mathsf{eval}(\mathsf{KB}, q_i).$$

Let $\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}$ *be such that*

$$\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_{i-1} = \mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_{i-1}$$

and

$$\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)_i = \mathsf{eval}(\mathsf{KB}, q_i).$$

Then it holds that

$$\mathsf{cont}(\mathsf{Cens}'_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q), i) \models \Box s \quad \textit{for some } s \in \mathsf{Sec}.$$

Next we define the notion of a repudiating censor, which garantees that there is always a knowledge base in which no secret holds and which, given as input to the answering function, produces the same results as the actual knowledge base. Hence this definition provides a version of plausible deniability for all secrets.

Definition 3.11. *A censor* Cens *is called* repudiating *iff for each privacy configuration* $(\mathsf{KB}, \mathsf{AK}, \mathsf{Sec})$ *and each query sequence* q*, there are knowledge bases* KB_i *(*$i \in \omega$*) such that*

1. $(\mathsf{KB}_i, \mathsf{AK}, \mathsf{Sec})$ *is a privacy configuration for each* $i \in \omega$*;*
2. $\mathsf{Cens}_{(\mathsf{KB},\mathsf{AK},\mathsf{Sec})}(q)|_n = \mathsf{Cens}_{(\mathsf{KB}_n,\mathsf{AK},\mathsf{Sec})}|_n$*, for each* $n \in \omega$*;*
3. $\mathsf{KB}_i \not\models_{\mathsf{L}} s$ *for each* $s \in \mathsf{Sec}$ *and each* $i \in \omega$*.*

Now we can establish our first no-go theorem, which is a generalization of Th. 50 in Ref. Studer and Werner (2014).

Theorem 3.1 (First No-Go Theorem)**.** *A continuous and truthful censor satisfies at most two of the properties effectiveness, minimal invasion, and repudiation.*

4 NON-REFUSING CENSORS

In this section we study censors that do not refuse to answer a query.

Definition 4.1. *A censor is* non-refusing *if it never assigns the answer r to a query.*

Of course, a non-refusing censor has to lie in order to keep the secrets. That means if a censors of this kind shall be effective, then it cannot be truthful.

Even if we consider lying censors, we work with the assumption that

$$\text{an attacker believes every answer of the censor.} \tag{1}$$

Otherwise, we are in a situation where an attacker cannot believe any answer because the attacker does not know which answers are correct and which are wrong, which means that any answer could be a lie. In that case, querying a knowledge base would not make any sense at all.[1]

Because of the assumption (1), we can use our notions of effectiveness (Definition 3.9) and credibility (Definition 3.7) also in the context of lying censors: an attacker should not believe any secret and the beliefs should be satisfiable.

Theorem 3.1 about truthful censors did not make any assumptions on the underlying logic L. The next theorem about non-refusing censors is less general as it is based on classical logic. We will use $a, b, c, \ldots$ for atomic propositions and $A, B, C, \ldots$ for arbitrary formulas.

Moreover, we assume that the knowledge base KB only contains atomic facts (we say KB is *atomic*). That is if $F \in \mathsf{KB}$, then F is either of the form p or of the form $\neg p$ where p is an atomic proposition. Hence we find that if $\mathsf{KB} \vdash_{\mathsf{L}} a \to b$ for two distinct atomic propositions a and b, then $\mathsf{KB} \vdash_{\mathsf{L}} \neg a$ or $\mathsf{KB} \vdash_{\mathsf{L}} b$. We can formalize this using the set of a priori knowledge by letting

$$\Box(a \to b) \to (\Box\neg a \lor \Box b) \in \mathsf{AK}.$$

Now we can establish our second no-go theorem, which is a generalization of the results of Biskup (2000).

Theorem 4.1 (Second No-Go Theorem)**.** *Let L be based on classical logic. A continuous and non-refusing censor cannot be at the same time effective and minimally invasive.*

To avoid this problem, a censor must not only protect the single elements of Sec but also their disjunction (Biskup, 2000). Note that protecting the disjunction of all secrets is not as simple as it sounds. Consider, for instance, a hospital information system that should protect the disease a patient is diagnosed with. In this case, protecting the disjunction of all secrets means protecting the information that the patient has some disease. This, however, is not feasible as it is general background knowledge that everybody who is a patient in a hospital has some disease. Worse than that, sometimes the disjunction of all secrets may even be a logical tautology, which cannot be protected.

[1] This is, of course, not completely true. It is possible to distort knowledge bases in such a way that privacy is preserved but statistical inferences are still informative, see, e.g. Ref. du Pin Calmon and Fawaz (2012).

5 CONCLUSION

In this paper, we have established two no-go theorems for data privacy using tools from modal logic. We are confident that logical methods will play an important role for finding new impossibility theorems or for better understanding already known ones, see, e.g., the logical analyses carried out by Nurgalieva and del Rio (2019) and Pacuit and Yang (2016).

Another line of future research relates to the fact that refusing to answer a query can give away the information that there exists a secret that could be infered from some other answer. Similar phenomena may occur in multi-agent systems when one of the agents refuses to communicate. For example, imagine the situation of an oral exam where the examiner asks a question and the student keeps silent. In this case the examiner learns that the student does not know the answer to the question for otherwise the student would have answered.

It is also possible that refusing an answer can lead to knowing that someone else knows a certain fact. Consider the following scenario. A father enters a room where his daughter is playing and he notices that one of the toys is in pieces. So he asks who has broken the toy. The daughter does not want to betray her brother (who actually broke it) and she also does not want to lie. Therefore, she refuses to answer her father's question. Of course, then the father knows that his daughter knows who broke the toy for otherwise the daughter could have said that she does not know.

We believe that it is worthwhile to study the above situations using general communication protocols that include the possibility of refusing an answer and to investigate the implications of refusing in terms of higher-order knowledge.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers, who helped to improve this paper substantially. This work is supported by the Swiss National Science Foundation grant 200020_184625.

REFERENCES

Ågotnes, T., van Ditmarsch, H., and Wang, Y. (2018). True lies. *Synthese*, 195(10):4581–4615.

Arrow, K. J. (1950). A difficulty in the concept of social welfare. *Journal of Political Economy*, 58(4):328–346.

Avron, A. (1991). Simple consequence relations. *Inf. Comput.*, 92(1):105–139.

Bell, J. S. (1964). On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1:195–200.

Biskup, J. (2000). For unknown secrecies refusal is better than lying. *Data and Knowledge Engineering*, 33(1):1–23.

Biskup, J. and Bonatti, P. A. (2001). Lying versus refusal for known potential secrets. *Data and Knowledge Engineering*, 38(2):199–222.

Biskup, J. and Bonatti, P. A. (2004a). Controlled query evaluation for enforcing confidentiality in complete information systems. *International Journal of Information Security*, 3(1):14–27.

Biskup, J. and Bonatti, P. A. (2004b). Controlled query evaluation for known policies by combining lying and refusal. *Annals of Mathematics and Artificial Intelligence*, 40(1):37–62.

Biskup, J. and Weibert, T. (2008). Keeping secrets in incomplete databases. *International Journal of Information Security*, 7(3):199–217.

Bonatti, P. A., Kraus, S., and Subrahmanian, V. S. (1995). Foundations of secure deductive databases. *Transactions on Knowledge and Data Engineering*, 7(3):406–422.

du Pin Calmon, F. and Fawaz, N. (2012). Privacy against statistical inference. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1401–1408. IEEE.

Frauchiger, D. and Renner, R. (2018). Quantum theory cannot consistently describe the use of itself. *Nature Communications*, 9.

Icard, B. (2019). *Lying, deception and strategic omission : definition et evaluation*. PhD thesis, Universit Paris Sciences et Lettres.

Iemhoff, R. (2016). Consequence relations and admissible rules. *Journal of Philosophical Logic*, 45(3):327–348.

Kochen, S. and Specker, E. (1968). The problem of hidden variables in quantum mechanics. *Indiana Univ. Math. J.*, 17:59–87.

Nurgalieva, N. and del Rio, L. (2019). Inadequacy of modal logic in quantum settings. In Selinger, P. and Chiribella, G., editors, *Proceedings 15th International Conference on Quantum Physics and Logic, QPL 2018, Halifax, Canada, 3-7th June 2018.*, volume 287 of *EPTCS*, pages 267–297.

Pacuit, E. and Yang, F. (2016). Dependence and independence in social choice: Arrow's theorem. In Abramsky, S., Kontinen, J., Väänänen, J., and Vollmer, H., editors, *Dependence Logic: Theory and Applications*, pages 235–260. Springer.

Sicherman, G. L., De Jonge, W., and Van de Riet, R. P. (1983). Answering queries without revealing secrets. *ACM Trans. Database Syst.*, 8(1):41–59.

Stoffel, K. and Studer, T. (2005). Provable data privacy. In Andersen, K. V., Debenham, J., and Wagner, R., editors, *Database and Expert Systems Applications*, pages 324–332. Springer.

Stouppa, P. and Studer, T. (2007). A formal model of data privacy. In Virbitskaite, I. and Voronkov, A., editors, *Perspectives of Systems Informatics*, pages 400–408. Springer.

Studer, T. (2020). No-go theorems for data privacy. E-print 2005.13811, arXiv.org.

Studer, T. and Werner, J. (2014). Censors for boolean description logic. *Transactions on Data Privacy*, 7:223–252.

van Ditmarsch, H. (2014). Dynamics of lying. *Synthese*, 191(5):745–777.

Hill Cipher Key Generation Using Skew-symmetric Matrix

Khang Jie Liew[*1] and **Van Thai Nguyen**[1]

[1]*Centre for American Education, Sunway University*

E-mail: khangjiel@sunway.edu.my
[*]*Corresponding author*

ABSTRACT

Hill cipher refers to a symmetric cryptosystem with several advantages in transmitting a secret message. It appears to be the first polygraphic cipher that adopts the concept of linear algebra by performing its operations on a square matrix to both encrypt and decrypt messages. The Hill cipher, nonetheless, has been associated with the following issues: A square matrix that serves as the key matrix must be invertible to allow ciphertext decryption. Next, the Hill cipher is susceptible to known-plaintext attack due to its linear nature. The main contribution of this paper is to propose a secure variant of affine Hill cipher to overcome the mentioned issues above. A skew-symmetric matrix is transformed into an invertible orthogonal matrix to function as the key matrix. Next, a random sequence of vectors is generated by a fixed seed number as the additional step for the encryption process. All operations are performed under modular arithmetic. The proposed variant successfully decreased computational time in the decryption process. A detailed example is provided to illustrate the proposed algorithm.

Keywords: Hill cipher, skew symmetric matrix, orthogonal matrix, key matrix, seed number, encryption, decryption

1 INTRODUCTION

In this information age, humans communicate and exchange information via electronic form. This highlights the significance of electronic security to hinder information from being attacked, stolen, or altered by irresponsible third-party. That being said, cryptography plays an integral role in providing effective techniques to enhance the security of information transmission. Cryptography refers to the study of methods of sending messages in disguised form over an insecure channel in such a way that only the intended receiver can remove the disguise and read the message (Koblitz, 1987). This has been widely applied in email communication, e-payment, electronic voting, and automated teller machine, to name a few. At present time, cryptography is considered as a branch of mathematics and computer science. The earliest known cryptosystem was introduced by the Roman ruler, Julius Caesar. He employed simple cipher to communicate in secret, hence known as Caesar's cipher. This cipher reflects the concept of substitution and has been considered as the simplest encryption method. Generally, the two types of cryptosystems are symmetric and asymmetric cryptosystems. In symmetric cryptosystem, both encryption and decryption keys are similar, while varied keys are used in asymmetric cryptosystem for encryption and decryption. Hill cipher, which was proposed in 1929 by a mathematician named Lester Hill, refers to a symmetric cryptosystem that heavily relies on its matrix operation. In precise, the Hill cipher is a monoalphabetic polygraphic cipher; a cipher that applies fixed substitution for the whole message and operates on larger groups of letters.

Some advantages of Hill cipher are its ability to disguise as the letter frequencies of the plaintext, as well as the simple and rapid encryption and encryption processes. Meanwhile, the issues related to the Hill cipher as follows: It requires a key matrix that is invertible to allow decryption. It can be easily broken by known-plaintext attack. It is unsuitable for encryption of plaintext that contains all zero entries. To overcome the difficulty of known-plaintext attack, Kumar et al. (2006) had adopted an iterative method in their study. In order to overcome the problem of inverse key matrix, self-invertible matrix was proposed by Acharya et al. (2007) that succeeded in image encryption (Acharya et al., 2008a). Nonetheless, there are no mathematical proofs that self-invertible matrix is always invertible and it is time-consuming to construct a self-invertible matrix with higher dimension. In year 2008, Acharya et al. (2008b) proposed a technique that enables adjustments made to the key matrix to form a different key for each block of encryption while still using self-invertible matrix as the key. The three matrix keys generations are involutory, permuted, and reiterative matrix, as proposed by Acharya et al. (2009b), to address the issue of matrix inversion and to enhance the security of Hill system. The involutory key matrix encrypts images with homogenous background and this scheme is called advanced Hill cipher algorithm (Acharya et al., 2009a). However, the scheme does not solve the plaintext block with all zero entries. Toorani and Falahati (2009) extended the concept of Hill cipher to affine the Hill cipher and to increase its randomness, despite the possibility that the key matrix is non-invertible. Rahman et al. (2013) proposed another variant of affine Hill cipher to be applied in the involuntary matrix as the key matrix. The authors introduced three parameters as the additional secret keys. The proposed method enhanced the randomness of the existing method and addressed the mentioned issue, except that the parameters increased the system intricacy. Sharma and Chirgaiya (2014) initiated a new variant of Hill cipher to both encrypt and decrypt messages despite the non-invertible key matrix. However, the proposed method was unjustified due to limitation in mathematical proofs. Several other specific matrices, such as Vandermonde matrix (Sharma and Rehan, 2014) and orthogonal matrix (Khan et al., 2015), have been proposed as key matrices. The applications of Hill cipher are not limited to cryptography but can also be applied to transform biometric signals (Kaur and Khanna, 2017).

Turning to this study, a secure variant of affine Hill cipher is proposed. The variant is meant to address issues of invertible key matrix, namely known-plaintext attack and all zeros plaintext block. The key matrix generation was initiated from a skew-symmetric matrix with random integer entries. Next, algebraic operations were incorporated into the skew-symmetric matrix to generate an orthogonal matrix. The inverse of key matrix was easily obtained by transposing the orthogonal matrix. A random sequence of vectors produced from a fixed seed number was embedded into the product of key matrix and plaintext. The secret keys shared between sender and receiver are the key matrix and the seed number. All operations were performed under modular arithmetic. The proposed variant reduced the time to calculate the inverse of key matrix and addressed algorithm intricacy.

The mathematical background, such as the existing Hill cipher, matrix operation, modular arithmetic, skew-symmetric matrix, and orthogonal matrix, is briefly described in Section 2, along with mathematical proofs. Section 3 presents the proposed encryption and decryption algorithms, including an example to illustrate the algorithms. Lastly, the study is concluded in Section 4.

2 MATERIAL AND METHODS

2.1 Hill Cipher

The Hill cipher employs an area of mathematics called linear algebra developed by a mathematician, Lester Hill. Prior to encryption, the plaintext is divided into a few blocks of letters, such as two letters per block (diagraphs), three letters per block (trigraph), or theoretically any block size. Next, each letter is represented by a number modulo with a positive integer n. If $n = 26$ is applied, each letter is substituted by a numerical value, such as $A = 0,\ B = 1,\ C = 2, \ldots,\ Z = 25$. Although this scheme is often used, it is not a standard way. If there are m letters per block or in precise, the column vector with m entries, then the invertible key matrix used for encryption is $m \times m$ square matrix K. Let $\mathbf{P}$ be the plaintext vector with size m. Note that the m numerical entries in the plaintext vector is equivalent to the m letters. Hence, the encryption is $\mathbf{C} = K\mathbf{P}$ (mod n). The ciphertext $\mathbf{C}$ refers to the column vector with size m obtained by using the method of linear transformation. During decryption, the plaintext of $\mathbf{P}$ can be obtained, such that $\mathbf{P} = K^{-1}\mathbf{C}$ (mod n). The decryption is unique when K^{-1} exists in modulo n. This can only be satisfied if gcd(det K(mod n), n)= 1. A prime number n is employed to hinder the determinant of key matrix from having common factor with modulo n. This is bound to happen when n is a composite. This study assessed the affine Hill cipher by extending its concept. It has the form of $\mathbf{C} = K\mathbf{P} + \mathbf{V}$, where $\mathbf{V}$ denotes column vector with similar size of vectors $\mathbf{C}$ and $\mathbf{P}$.

2.2 Modular Arithmetic

Modular arithmetic refers to the study of arithmetic with congruence classes. It is briefly described to enhance the understanding of the coming section. The proofs are omitted as this subsection is taken from Jones and Jones (2012).

Definition 2.1. *Let n be a positive integer, and let a and b be any integers. Then a is congruent to b, written $a \equiv b \pmod{n}$.*

Lemma 2.1. *For any fixed $n \geq 1$, $a \equiv b \pmod{n}$ if any only if $n|(a - b)$.*

Lemma 2.2. *For any fixed $n \geq 1$,*

1. *$a \equiv a \pmod{n}$ for all integers a;*
2. *if $a \equiv b \pmod{n}$, then $b \equiv a \pmod{n}$;*
3. *if $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, then if $a \equiv c \pmod{n}$.*

Lemma 2.3. *For a given $n \geq 1$, if $a' \equiv a \pmod{n}$ and $b' \equiv b \pmod{n}$, then $a' + b' \equiv a + b \pmod{n}$, $a' - b' \equiv a - b \pmod{n}$, and $a'b' \equiv ab \pmod{n}$.*

Definition 2.2. *A multiplicative inverse for a class $[a] \in \mathbb{Z}_n$ is a class $[b] \in \mathbb{Z}_n$ such that $[a][b] = [1]$. A class $[a] \in \mathbb{Z}_n$ is a unit if it has a multiplicative inverse in $\mathbb{Z}_n$. In other words, the integer a is a unit modulo n means that $ab \equiv 1$ (mod n) for some integer b.*

Lemma 2.4. *$[a]$ is unit in $\mathbb{Z}_n$ if and only if gcd(a,n)$= 1$.*

2.3 Matrices

Some elementary linear algebra concepts especially the skew-symmetric matrix and orthogonal matrix are being introduced in this subsection.

Definition 2.3. *(Kolman and Hill, 2008) An $m \times m$ matrix A is called nonsingular or invertible, if there exists $m \times m$ matrix B such that $AB = BA = I_m$; such a B is called an inverse of A and it is unique. Otherwise, A is called singular or non-invertible.*

Corollary 2.1. *If A is an $m \times m$ matrix, then A is nonsingular if and only if det$(A) \neq 0$.*

Definition 2.4. *(Babu, 2010) An $m \times m$ matrix $A = [a_{ij}]$ is said to be skew-symmetric matrix if $A^T = -A$, that is, $a_{ij} = -a_{ji}$ for all i and j.*

All the diagonal elements are zero in skew-symmetric matrix. For example, the 3×3 skew-symmetric matrix has the general form $\begin{bmatrix} 0 & -h & -g \\ h & 0 & -f \\ g & f & 0 \end{bmatrix}$, where $f,\ g\ , h$ are any nonzero real number.

To determine whether a skew-symmetric matrix invertible, the following argument is provided. Suppose that A is an $m \times m$ skew symmetric matrix with real entries. From the definition, $A^T = -A$, then det $(A^T) =$ det $(-A)$. This leads to the det $A = (-1)^m$det A. If m is odd, det $A = 0$, but it is inconclusive when m is even. However, if m is even and the entries of the skew-symmetric matrix are integer, then its determinant is a perfect square (Buontempo, 1982). The determinant is zero when m is odd implies that the skew-symmetric matrix is non-invertible and thus zero is one of the eigenvalues of skew-symmetric matrix. However, the nonzero eigenvalues exist but they are purely imaginary numbers bi, where $b \in \mathbb{R}\backslash\{0\}$.

Theorem 2.1. *If A is an $m \times m$ skew-symmetric matrix, then matrices $I + A$ and $I - A$ are invertible.*

Proof. Suppose that A is an $m \times m$ skew-symmetric matrix. Assume that a scalar λ is an eigenvalue of A that satisfies $A\mathbf{x} = \lambda\mathbf{x}$ for some nonzero vector $\mathbf{x}$. Also, the eigenvalue 1 satisfies $I\mathbf{x} = 1\mathbf{x}$ for some nonzero vector $\mathbf{x}$. Then $I\mathbf{x} + A\mathbf{x} = 1\mathbf{x} + \lambda\mathbf{x}$ which is $(I + A)\mathbf{x} = (1 + \lambda)\mathbf{x}$. Similarly, $I\mathbf{x} - A\mathbf{x} = 1\mathbf{x} - \lambda\mathbf{x}$ which is $(I - A)\mathbf{x} = (1 - \lambda)\mathbf{x}$. Thus, the eigenvalues of $I + A$ and $I - A$ are $(1 + \lambda)$ and $(1 - \lambda)$, respectively. From the fact that eigenvalues of A are zero or purely imaginary numbers bi, then eigenvalues $(1 \pm \lambda)$ are nonzero. This implies that the determinant of $I + A$ and $I - A$ are nonzero and hence these two matrices are invertible. □

Definition 2.5. *(Kolman and Hill, 2008) An $m \times m$ matrix A is called orthogonal if $A^{-1} = A^T$. In another words, A is orthogonal if $A^T A = I_m$.*

Theorem 2.2. *Let A and I be an $m \times m$ matrix and identity matrix, respectively. The matrices $(I + A)$ and $(I - A)$ are commutative.*

Proof. Suppose that A is an $m \times m$ matrix and I is an identity matrix. Now $(I + A)(I - A) = (I + A)I - (I + A)A = I^2 + AI - AI - A^2 = I^2 - A^2 = I - A^2$ and similarly , $(I - A)(I + A) = (I - A)I + (I - A)A = I^2 - AI + AI - A^2 = I^2 - A^2 = I - A^2$. Hence, matrices $(I + A)$ and $(I - A)$ commute. □

Theorem 2.3. *Let A and I be an $m \times m$ skew-symmetric matrix and identity matrix, respectively. The matrix $(I - A)(I + A)^{-1}$ is an orthogonal matrix.*

Proof. Suppose that A is an $m \times m$ skew-symmetric matrix and I is an identity matrix. If $(I - A)(I + A)^{-1}$ is an orthogonal matrix, then $((I - A)(I + A)^{-1})^T((I - A)(I + A)^{-1}) = I$.

$$\begin{aligned}
& ((I - A)(I + A)^{-1})^T((I - A)(I + A)^{-1}) \\
= \; & ((I + A)^{-1})^T(I - A)^T((I - A)(I + A)^{-1}) \\
= \; & ((I + A)^T)^{-1}(I - A)^T((I - A)(I + A)^{-1}) \\
= \; & ((I + A^T))^{-1}(I - A^T)((I - A)(I + A)^{-1}) \\
= \; & ((I - A)^{-1}(I + A))((I - A)(I + A)^{-1}) \\
= \; & (I - A)^{-1}((I + A)(I - A))(I + A)^{-1} \\
= \; & ((I - A)^{-1}(I - A))((I + A)(I + A)^{-1}) \\
= \; & I \cdot I \qquad \text{(from Theorem 6)} \\
= \; & I
\end{aligned}$$

Therefore, the matrix $(I - A)(I + A)^{-1}$ is an orthogonal matrix. □

After unravelling several important concepts, the following section discusses the proposed algorithms. A concrete example is embedded to attain better understanding.

3 RESULTS AND DISCUSSION

In order to embed plaintext via Hill cipher algorithm, the 26 capital letters in English were assigned to various numerical values, as depicted in the earlier section. Three symbols, namely ",", "_", and "." were represented by 26, 27, and 28, respectively, so that modulo $n = 29$, which is a prime, could be considered throughout the algorithm. A plaintext that contained m letters was assigned to the entry of $r \times m$ matrix E. The value of r reflects the number of blocks determined by the ceiling function, such that $r = \lceil \frac{L}{m} \rceil$, where L is the length of letters and m

denotes the desired entries per block. When the last few entries of matrix E were not filled, a dummy letter X was used. Next, the letters in the entries of matrix E were substituted by the numerical values assigned earlier. In this proposed algorithm, both sender and receiver share similar key matrix and seed number. Further details regarding the key generation, proposed encryption, and decryption algorithms are given in the following.

Algorithm 1 The Key Generation Algorithm for the Proposed Hill Cipher

```
Generate the $m \times m$ skew-symmetric matrix $A$ with integer entries randomly.
Compute the key matrix $K = (I - A)(I + A)^{-1} (\text{mod}\, n)$.
```

The mathematical proofs of the key generation technique are available in subsection 2.3. The next algorithms 2 and 3 are the encryption and decryption algorithms for the proposed Hill cipher.

Algorithm 2 The Encryption Algorithm for the Proposed Hill Cipher

```
Determine the number of blocks of plaintext $r$.
Substitute the plaintext by numerical values.
Assign the numerical plaintext to the $r \times m$ matrix.
Generate the key as in Algorithm 1.
Fix a seed number and generate the first $r$ terms of sequence of random vector, $V$, with integer entries.
for $i =$ plaintext vector 1 : plaintext vector $r$ do
    Encrypt the plaintext, such that $\mathbf{C}_i = K\mathbf{P}_i + \mathbf{v}_i$ (mod $n$), where $C$ and $P$ are ciphertext and plaintext, respectively.
end for
The numerical ciphertext is converted to letter ciphertext and sent.
```

Algorithm 3 The Decryption Algorithm for the Proposed Hill Cipher

```
Receive and convert the letter ciphertext to numerical ciphertext.
Calculate the inverse of key matrix $K$ (mod $n$). Matrix $K$ is an orthogonal matrix, while the inverse is $K^{-1} = K^T$.
Use the agreed seed number to generate the first $r$ term of sequence of random vector, $V$ with integer entries.
for $i =$ ciphertext vector 1 : ciphertext vector $r$ do
    Encrypt the ciphertext, such that $\mathbf{P}_i = K^{-1}(\mathbf{C}_i - \mathbf{v}_i)$ (mod $n$).
end for
Convert the numerical plaintext to letter plaintext.
```

The efficiency of Algorithms 2 and 3 is $O(r + m^3 + rm)$ and $O(r + rm)$, respectively. It is obvious that the computational time for the decryption process is shorter than the encryption process. The following is a numerical instance to illustrate the implementation of the proposed algorithm using Eclipse Java 2019-06. Here, all operations were performed under modulo 29 by using Trigraph Hill cipher.

Key Generation and Encryption:

Suppose the sender, Alice, wishes to send the message "ATTACK" through an insecure channel. She substitutes the plaintext with numerical values and gets $\{0, 19, 19, 0, 2, 10\}$. The length of the letters is $L = 6$ and the desired $m = 3$, so the required blocks are $r = 2$. Next, the numerical plaintext is converted to 2×3 matrix E, such that $E = \begin{bmatrix} 0 & 19 & 19 \\ 0 & 2 & 10 \end{bmatrix}$. The first and second rows of the matrix refer to the first and second blocks of the Hill cipher. Alice generates a random skew-symmetric matrix with integer entries. The skew-symmetric matrix is $A = \begin{bmatrix} 0 & 377 & 73 \\ -377 & 0 & -227 \\ -73 & 227 & 0 \end{bmatrix}$. Next, the key matrix $K = (I - A)(1 + A)^{-1} \pmod{29}$ is calculated. The key matrix is $K = \begin{bmatrix} 17 & 15 & 3 \\ 15 & 6 & 1 \\ 26 & 28 & 22 \end{bmatrix}$. Alice encrypts the first and second blocks using the key matrix K under mod 29. She obtains $\begin{bmatrix} 23 \\ 17 \\ 22 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 22 \\ 15 \end{bmatrix}$. Alice fixes the seed number 21 and generates a sequence of random vector that is composed of three integer entries. The fixed seed is shared between Alice and Bob. The first two vectors, $\mathbf{v}_1$ and $\mathbf{v}_1$, are $\begin{bmatrix} 724 \\ 320 \\ 627 \end{bmatrix}$ and $\begin{bmatrix} -746 \\ -824 \\ -352 \end{bmatrix}$, which are added to $\begin{bmatrix} 23 \\ 17 \\ 22 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 22 \\ 15 \end{bmatrix}$. The sum of the two vectors is carried out under modulo 29. The required numerical ciphertext is $\begin{bmatrix} 22 \\ 18 \\ 11 \end{bmatrix}$ and $\begin{bmatrix} 10 \\ 10 \\ 11 \end{bmatrix}$. After that, the numerical ciphertext is changed to letter ciphertext, which results in "CWPWSL". Note that the randomness in generating the sequence of random vector is able to protect the ciphertext from known-plaintext attack.

Decryption:

Bob receives the ciphertext and converts it to numerical ciphertext, which is $\begin{bmatrix} 22 \\ 18 \\ 11 \end{bmatrix}$ and $\begin{bmatrix} 10 \\ 10 \\ 11 \end{bmatrix}$. The inverse of key matrix is obtained by transposing the key matrix K, which refers to $K^{-1} = \begin{bmatrix} 17 & 15 & 26 \\ 15 & 6 & 28 \\ 3 & 1 & 22 \end{bmatrix}$. Bob also generates a sequence of random vector that has three integer entries using the agreed seed number. He encrypts the ciphertext and gets the numerical plaintext through

$$\begin{bmatrix} 17 & 15 & 26 \\ 15 & 6 & 28 \\ 3 & 1 & 22 \end{bmatrix} \left(\begin{bmatrix} 22 \\ 18 \\ 11 \end{bmatrix} - \begin{bmatrix} 724 \\ 320 \\ 627 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 19 \\ 19 \end{bmatrix} \pmod{29}$$

and

$$\begin{bmatrix} 17 & 15 & 26 \\ 15 & 6 & 28 \\ 3 & 1 & 22 \end{bmatrix} \left(\begin{bmatrix} 10 \\ 10 \\ 11 \end{bmatrix} - \begin{bmatrix} -746 \\ -824 \\ -352 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 2 \\ 10 \end{bmatrix} \pmod{29}.$$

The numerical plaintext is converted to letter plaintext, which reveals "ATTACK", as desired.

4 CONCLUSION

The proposed algorithm is a secure variant of affine Hill cipher. It is based on the skew-symmetric matrix, which is transformed into an orthogonal matrix and a sequence of random vector by using a fixed seed number. The use of the orthogonal matrix as the key matrix appears to be efficient as its inverse could be easily obtained, thus reducing computational time for the decryption process. The addition of random vectors enhances the security of the cipher by preventing the known-plaintext attack. The efficiency of the algorithms, nevertheless, could be affected due to lengthy messages. The future works are to determine the suitability of proposed algorithms to encrypt the plaintext that contains all zero entries and comparative analysis with the existing Hill cipher variant.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Dr. Wen Chean Teh for in-depth linear algebra discussion and Ms. Tee Hean Tan for her invaluable aid in Java programming.

REFERENCES

Acharya, B., Panigrahy, S. K., Patra, S. K., and Panda, G. (2009a). Image encryption using advanced hill cipher algorithm. *International Journal of Recent Trends in Engineering*, 1(1):663–667.

Acharya, B., Patra, S. K., and Panda, G. (2008a). Image encryption by novel cryptosystem using matrix transformation. In *Proceedings of First International Conference on Emerging Trends in Engineering and Technology 2008*, pages 77–81. IEEE.

Acharya, B., Patra, S. K., and Panda, G. (2009b). Involutory, permuted and reiterative key matrix generation methods for hill cipher system. *International Journal of Recent Trends in Engineering*, 1(4):106–108.

Acharya, B., Rath, G. S., and Patra, S. K. (2008b). Novel modified hill cipher algorithm. In *Proceedings of the International Conference on Emerging Technologies and Applications in Engineering Technology and Sciences 2008*, pages 126–130.

Acharya, B., Rath, G. S., Patra, S. K., and Panigrahy, S. K. (2007). Novel methods of generating self-invertible matrix for hill cipher algorithm. *International Journal of Security*, 1(1):14.

Babu, R. (2010). *Engineering Mathematics I: For Uptu*. Pearson Education.

Buontempo, D. J. (1982). The determinant of a skew-symmetric matrix. *The Mathematical Gazette*, 66(435):67–69.

Jones, G. A. and Jones, J. M. (2012). *Elementary Number Theory*. Springer Undergraduate Mathematics Series. Springer London.

Kaur, H. and Khanna, P. (2017). Non-invertible biometric encryption to generate cancelable biometric templates. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 1–4.

Khan, F. H., Shams, R., Qazi, F., and Agha, D. (2015). Hill cipher key generation algorithm by using orthogonal matrix. *Proceedings International Journal of Innovative Science and Modern Engineering*, 3(3):5–7.

Koblitz, N. (1987). *A Course in Number Theory and Cryptography*. Springer-Verlag, New York.

Kolman, B. and Hill, D. (2008). *Elementary Linear Algebra with Applications*. Pearson.

Kumar, S. U., Sastry, V., and Babu, A. V. (2006). A block cipher basing upon a revisit to the feistel approach and the modular arithmetic inverse of a key matrix. *IAENG International Journal of Computer Science*, 32(4):386–394.

Rahman, M. N. A., Abidin, A., Yusof, M. K., and Usop, N. (2013). Cryptography: a new approach of classical hill cipher. *International Journal of Security and Its Applications*, 7(2):179–190.

Sharma, N. and Chirgaiya, S. (2014). A novel approach to hill cipher. *International Journal of Computer Applications*, 108(11):34–37.

Sharma, P. and Rehan, M. (2014). Modified hill cipher using vandermonde matrix and finite field. *International Journal of Technology*, 4(1):252–256.

Toorani, M. and Falahati, A. (2009). A secure variant of the hill cipher. In *IEEE Symposium on Computers and Communications 2009*, pages 313–316. IEEE.

A Megabit Block Cipher

Nur Azman Abu[*1]

[1]*Faculty of ICT, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia*

E-mail: nura@utem.edu.my
[*]*Corresponding author*

ABSTRACT

An Advanced Encryption Standard(AES) has been the most popular block cipher in the last two decades. It has been extensively analyzed and efficiently implemented. Since 2000, an AES has been preset to be upgradable from the current 128-bit key to 192-bit key and finally 256-bit key on the same 128-bit plain text-cipher text block. A new call for 256-bit standard symmetric cipher is expected by 2030. Currently, an input file runs in kilobytes. It is apparent that a more practical cipher is much needed in handling daily task of protecting an important document from a user stand point of view without having to go through technical knowledge of encryption. A symmetric cipher has been traditionally operated on a small block. In this paper, however, a new proposal on a large 2048-bit block cipher using 256-bit key is presented.

Keywords: block cipher, Advanced Encryption Standard, S-box, P-box.

1 INTRODUCTION

Following a classical concept of confusion and diffusion, an element of the cipher is represented by S-box and P-box respectively. In this paper, a new proposal of 256-bit key on a large 2048-bit block cipher is presented. From the 256-bit key, 3 round keys will be generated. A block cipher with a higher number of rounds in a block cipher is expected to produce better crunching effect. In this cipher, there will be only two rounds with an S and P-boxes in each round. At the same time, a mix column transformation will be invoked immediately right after each S or P-box. This megabit cipher requires at least two S-boxes and two P-boxes.

Previously, an encryption on a large plaintext input file will be done via a chain codebook. A large input will be divided into blocks of n-bits and encrypted sequentially one block at a time. In AES, 128-bit block will be encrypted sequentially one block at a time. In this case, a cipher block size is 2048 bit (Rijmen and Daemen, 2001). A key on a block should be kept running

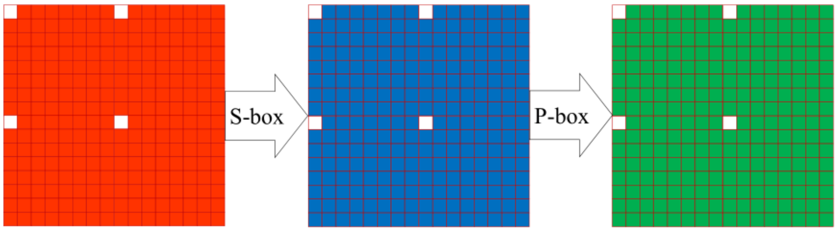


Figure 1: A 16×16 state array of bytes is compatible to any 16×16 S-box or P-box

continuously. There should be no easy way to differentiate a running transmission whether it is just a key stream or it carries a ciphertext or not.

In this proposal, a large 2048-bit block shall be first presented as 16 by 16 state byte array. A state array represents the plaintext and is being processed to get the ciphertext. The state array will go through a process of byte substitution and permutation via S-box and P-box respectively as shown in Figure 1.

From a classical concept of substitution and transposition, an element of a cipher is represented by S-box and P-box respectively. Nevertheless, they are insufficient to produce an exhaustive avalanche effect. Thus, a distinct mix column transformation will be invoked immediately right after each S or P-box in each round.

While AES is employing an irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ with $m(2) = 283_{10}$, candidates of irreducible polynomials to be used in this instance are listed in Table 1. In this mega cipher, several irreducible polynomials will be employed.

i	$m(x)$	Binary Coefficients	Decimal value $m_i(2)$
0	$x^8 + x^4 + x^3 + x + 1$	100011011_2	283_{10}
1	$x^8 + x^5 + x^3 + x + 1$	100101011_2	299_{10}
2	$x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$	101011111_2	351_{10}
3	$x^8 + x^6 + x^5 + x^3 + x + 1$	101100011_2	355_{10}
4	$x^8 + x^6 + x^5 + x^3 + x^2 + 1$	101100101_2	357_{10}
5	$x^8 + x^6 + x^5 + x^3 + x^3 + 1$	101101001_2	361_{10}
6	$x^8 + x^7 + x^6 + x + 1$	111000011_2	451_{10}
7	$x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$	111100111_2	487_{10}

Table 1: Candidates of irreducible polynomials to be used in this cipher

Traditionally, a session key will be used to pseudo-randomly generate round keys as a one-way process. A simple approach on a popular secure hashing algorithm may be called for here. Since the two round keys are coming from 24 consecutive hashing processes, the security of this cipher relies heavily on the strength of one-way function of SHA256.

2 ADVANCED ENCRYPTION STANDARD (AES)

Advanced Encryption Standard (AES) was specifically called for to replace the aging Data Encryption Standard (DES). Its selection procedure began back in (1997) by National Institute of Standards and Technology (NIST). When NIST summoned world's finest minds in the field of cryptography to cooperate by presenting their ideas for a new 128-bit encryption algorithm, RijnDael Algorithm was selected in the year 2000 (Rijmen and Daemen, 2001).

Prior to the year 2000, an AES has been preset to be upgradable from the current 128-bit key to 192-bit key and finally 256-bit key on the same 128-bit plain text-cipher text block. However, an increase in the bit length of the key size while maintaining the block size will not increase full complexity the exhaustive brute force attack on the plaintext block. A new call for 256-bit standard symmetric cipher is expected by 2030.

AES consists of ten rounds of basic operations, namely, S-box, shift row, mixed column and exclusive-or with round keys. In order to achieve a full collusion, AES takes 4 rounds of operations. An efficient AES implementation will combine 4 rounds of operations into one large lookup table to be exclusive-ored with 4 round keys at once. The whole process of encryption or decryption will be cut down to 4 and one half rounds.

3 OPTIMIZATION OF AES CIPHER

A direct comparison with an equivalent version of AES by Gladman (2002)'s original implementation in C language has been made. Using 32-bit processor, it is possible to speed up execution of this cipher by combining the SubBytes and ShiftRows steps with the MixColumns step by transforming them into a sequence of table lookups. This combination requires four 256-entry 32-bit tables (together occupying 4096 bytes). A round can then be performed with 16 table lookup operations and twelve 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step (Bertoni et al., 2002). Alternatively, the table lookup operation can be performed with a single 256-entry 32-bit table (occupying 1024 bytes) followed by circular rotation operations.

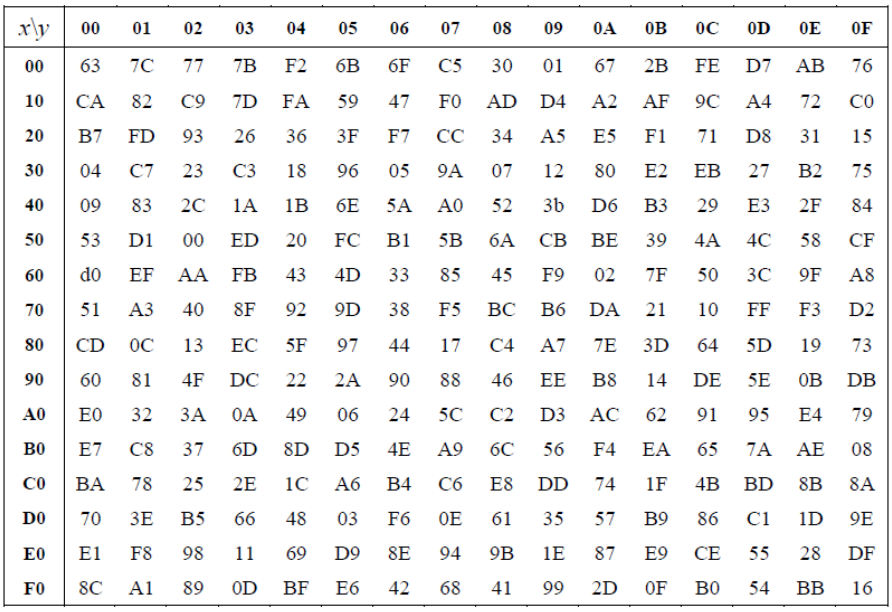
By 2030 in the next call for standard cipher, a block and key is expected to be 256 bit. In this proposal, a large 2048-bit block is presented. Nevertheless, a 256-bit plaintext may also be feed into the large 2048-bit block. As such pseudo randomly generated Round Key 0 from a session key should be random enough to disguise many zeros within a plaintext file.

3.1 Substitution Bytes

An S-box maps a byte x into output byte, $y = S(x)$. Both the input and output are interpreted as polynomials over GF(2). A substitution byte starts from converting an input byte into a polynomial over GF(2^8). An inverse of this polynomial will become an input to go through an affine transformation. This mathematical arena is well known to have good non-linearity properties

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Figure 2: An Affine Transformation in AES S-box



$x\backslash y$	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
10	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
20	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
30	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
40	09	83	2C	1A	1B	6E	5A	A0	52	3b	D6	B3	29	E3	2F	84
50	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
60	d0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
70	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
80	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
90	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A0	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B0	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C0	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D0	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E0	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F0	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

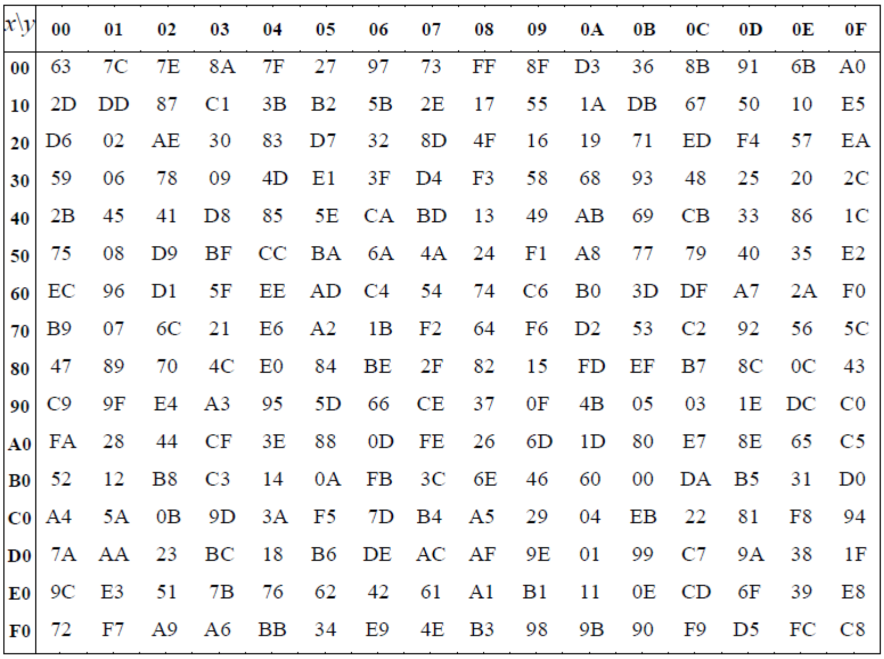
Table 2: An AES S-box in hexadecimal notation.

by applying an affine transformation to avoid attacks based on simple algebraic properties. The S-box is also used to avoid fixed points and opposite fixed points. An irreducible polynomial in AES is given by $m(x) = x^8 + x^4 + x^3 + x + 1$, or by $m(2) = 283_{10}$. Following similar technique, many more S-boxes can be generated from irreducible polynomials in Table 2.

3.2 Affine Transformation

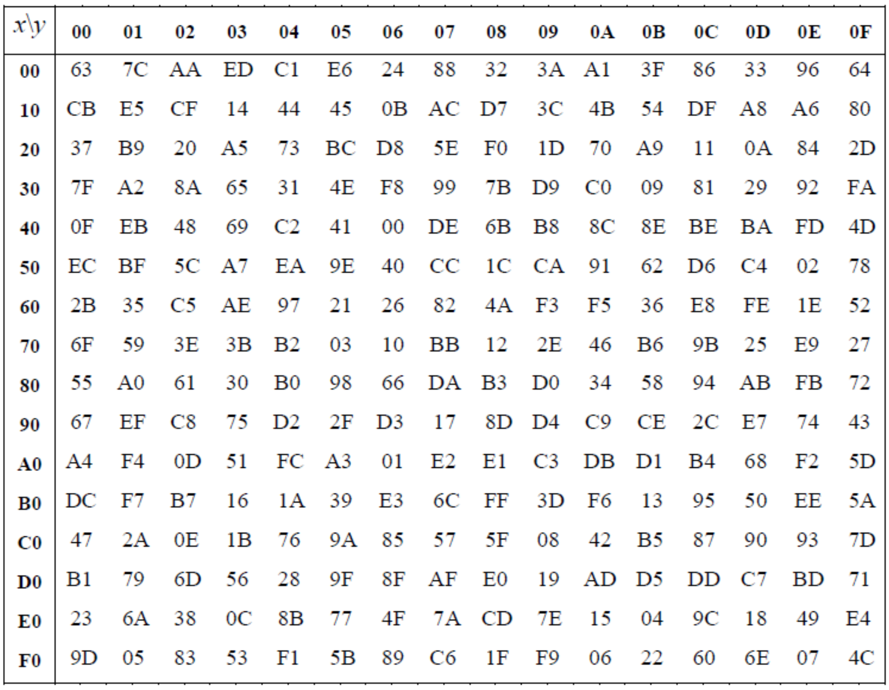
An affine transformation is a polynomial multiplication modulo a predefined irreducible polynomial. An affine transform can be compactly presented here by $y = Ax \oplus b \pmod{m(x)}$, where A is a constant matrix of 8×8 bits, x represents the value to transform when b is a constant byte equal to 6316 = 011000112 (Daemen and Rijmen, 2002). From an affine transform as shown in Figure 2, it is possible to construct different S-boxes using different matrix A which is a non-singular matrix, b and irreducible polynomials $m(x)$.

An AES S-box in hexadecimal notation is given in Table 2 is the byte substitution modulo an irreducible polynomial $m_0(x)$ which carry the value $m_0(2) = 283_{10}$. Following similar technique, four more S-boxes have been generated from irreducible polynomials $m_1(x)$, $m_2(x)$, $m_6(x)$ and $m_7(x)$. There are tabulated in Tables 3, 4, 5 and 6 respectively.



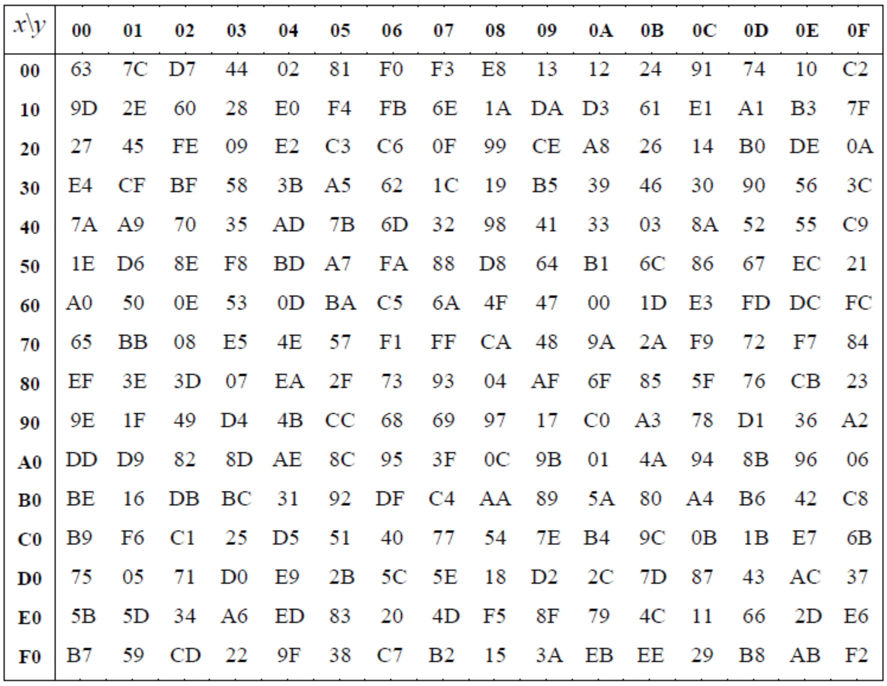
$x\backslash y$	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	7E	8A	7F	27	97	73	FF	8F	D3	36	8B	91	6B	A0
10	2D	DD	87	C1	3B	B2	5B	2E	17	55	1A	DB	67	50	10	E5
20	D6	02	AE	30	83	D7	32	8D	4F	16	19	71	ED	F4	57	EA
30	59	06	78	09	4D	E1	3F	D4	F3	58	68	93	48	25	20	2C
40	2B	45	41	D8	85	5E	CA	BD	13	49	AB	69	CB	33	86	1C
50	75	08	D9	BF	CC	BA	6A	4A	24	F1	A8	77	79	40	35	E2
60	EC	96	D1	5F	EE	AD	C4	54	74	C6	B0	3D	DF	A7	2A	F0
70	B9	07	6C	21	E6	A2	1B	F2	64	F6	D2	53	C2	92	56	5C
80	47	89	70	4C	E0	84	BE	2F	82	15	FD	EF	B7	8C	0C	43
90	C9	9F	E4	A3	95	5D	66	CE	37	0F	4B	05	03	1E	DC	C0
A0	FA	28	44	CF	3E	88	0D	FE	26	6D	1D	80	E7	8E	65	C5
B0	52	12	B8	C3	14	0A	FB	3C	6E	46	60	00	DA	B5	31	D0
C0	A4	5A	0B	9D	3A	F5	7D	B4	A5	29	04	EB	22	81	F8	94
D0	7A	AA	23	BC	18	B6	DE	AC	AF	9E	01	99	C7	9A	38	1F
E0	9C	E3	51	7B	76	62	42	61	A1	B1	11	0E	CD	6F	39	E8
F0	72	F7	A9	A6	BB	34	E9	4E	B3	98	9B	90	F9	D5	FC	C8

Table 3: S-box $1 \pmod{m_1(x)}$.



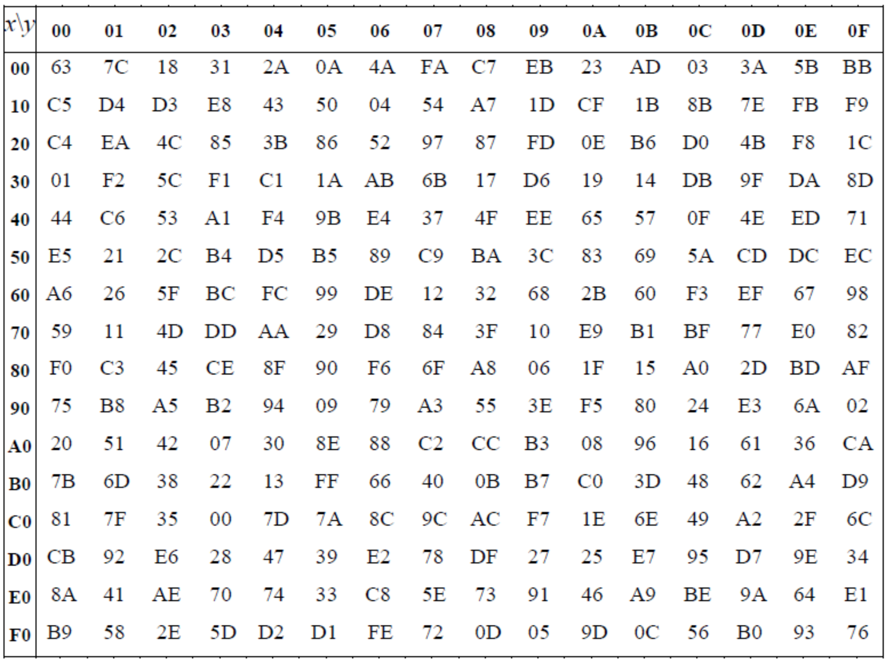
$x\backslash y$	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	AA	ED	C1	E6	24	88	32	3A	A1	3F	86	33	96	64
10	CB	E5	CF	14	44	45	0B	AC	D7	3C	4B	54	DF	A8	A6	80
20	37	B9	20	A5	73	BC	D8	5E	F0	1D	70	A9	11	0A	84	2D
30	7F	A2	8A	65	31	4E	F8	99	7B	D9	C0	09	81	29	92	FA
40	0F	EB	48	69	C2	41	00	DE	6B	B8	8C	8E	BE	BA	FD	4D
50	EC	BF	5C	A7	EA	9E	40	CC	1C	CA	91	62	D6	C4	02	78
60	2B	35	C5	AE	97	21	26	82	4A	F3	F5	36	E8	FE	1E	52
70	6F	59	3E	3B	B2	03	10	BB	12	2E	46	B6	9B	25	E9	27
80	55	A0	61	30	B0	98	66	DA	B3	D0	34	58	94	AB	FB	72
90	67	EF	C8	75	D2	2F	D3	17	8D	D4	C9	CE	2C	E7	74	43
A0	A4	F4	0D	51	FC	A3	01	E2	E1	C3	DB	D1	B4	68	F2	5D
B0	DC	F7	B7	16	1A	39	E3	6C	FF	3D	F6	13	95	50	EE	5A
C0	47	2A	0E	1B	76	9A	85	57	5F	08	42	B5	87	90	93	7D
D0	B1	79	6D	56	28	9F	8F	AF	E0	19	AD	D5	DD	C7	BD	71
E0	23	6A	38	0C	8B	77	4F	7A	CD	7E	15	04	9C	18	49	E4
F0	9D	05	83	53	F1	5B	89	C6	1F	F9	06	22	60	6E	07	4C

Table 4: S-box $1 \pmod{m_2(x)}$.



$x\backslash y$	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	D7	44	02	81	F0	F3	E8	13	12	24	91	74	10	C2
10	9D	2E	60	28	E0	F4	FB	6E	1A	DA	D3	61	E1	A1	B3	7F
20	27	45	FE	09	E2	C3	C6	0F	99	CE	A8	26	14	B0	DE	0A
30	E4	CF	BF	58	3B	A5	62	1C	19	B5	39	46	30	90	56	3C
40	7A	A9	70	35	AD	7B	6D	32	98	41	33	03	8A	52	55	C9
50	1E	D6	8E	F8	BD	A7	FA	88	D8	64	B1	6C	86	67	EC	21
60	A0	50	0E	53	0D	BA	C5	6A	4F	47	00	1D	E3	FD	DC	FC
70	65	BB	08	E5	4E	57	F1	FF	CA	48	9A	2A	F9	72	F7	84
80	EF	3E	3D	07	EA	2F	73	93	04	AF	6F	85	5F	76	CB	23
90	9E	1F	49	D4	4B	CC	68	69	97	17	C0	A3	78	D1	36	A2
A0	DD	D9	82	8D	AE	8C	95	3F	0C	9B	01	4A	94	8B	96	06
B0	BE	16	DB	BC	31	92	DF	C4	AA	89	5A	80	A4	B6	42	C8
C0	B9	F6	C1	25	D5	51	40	77	54	7E	B4	9C	0B	1B	E7	6B
D0	75	05	71	D0	E9	2B	5C	5E	18	D2	2C	7D	87	43	AC	37
E0	5B	5D	34	A6	ED	83	20	4D	F5	8F	79	4C	11	66	2D	E6
F0	B7	59	CD	22	9F	38	C7	B2	15	3A	EB	EE	29	B8	AB	F2

Table 5: S-box 2 $\pmod{m_6(x)}$.



$x\backslash y$	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	63	7C	18	31	2A	0A	4A	FA	C7	EB	23	AD	03	3A	5B	BB
10	C5	D4	D3	E8	43	50	04	54	A7	1D	CF	1B	8B	7E	FB	F9
20	C4	EA	4C	85	3B	86	52	97	87	FD	0E	B6	D0	4B	F8	1C
30	01	F2	5C	F1	C1	1A	AB	6B	17	D6	19	14	DB	9F	DA	8D
40	44	C6	53	A1	F4	9B	E4	37	4F	EE	65	57	0F	4E	ED	71
50	E5	21	2C	B4	D5	B5	89	C9	BA	3C	83	69	5A	CD	DC	EC
60	A6	26	5F	BC	FC	99	DE	12	32	68	2B	60	F3	EF	67	98
70	59	11	4D	DD	AA	29	D8	84	3F	10	E9	B1	BF	77	E0	82
80	F0	C3	45	CE	8F	90	F6	6F	A8	06	1F	15	A0	2D	BD	AF
90	75	B8	A5	B2	94	09	79	A3	55	3E	F5	80	24	E3	6A	02
A0	20	51	42	07	30	8E	88	C2	CC	B3	08	96	16	61	36	CA
B0	7B	6D	38	22	13	FF	66	40	0B	B7	C0	3D	48	62	A4	D9
C0	81	7F	35	00	7D	7A	8C	9C	AC	F7	1E	6E	49	A2	2F	6C
D0	CB	92	E6	28	47	39	E2	78	DF	27	25	E7	95	D7	9E	34
E0	8A	41	AE	70	74	33	C8	5E	73	91	46	A9	BE	9A	64	E1
F0	B9	58	2E	5D	D2	D1	FE	72	0D	05	9D	0C	56	B0	93	76

Table 6: S-box 2 $\pmod{m_7(x)}$.

3.3 Substitution bytes (S-box) and permutation bytes (P-box)

Confusion is an important concept in symmetric block cipher. A cipher needs to completely obscure statistical properties of an original message. A bit change in a plaintext should cause on average fifty percent changes in the final ciphertext. An S-box and P-box transform blocks of byte input into byte output. S-box is a key-less fixed substitution cipher and contains permutation of 256 bytes and 8-bit values. Each input byte is mapped or replaced with a corresponding new byte in S-box. From 0 to 255 byte values S-box is a one to one mapping to another byte. However, P-box is a permutation operation, without changing the output value, will change the index of the byte location.

In this instance, $m_1(x)$ and $m_2(x)$ have been selected as the irreducible polynomials to generate S-box 1 and P-box 1 respectively. At the same time, $m_6(x)$ and $m_7(x)$ have been selected as the irreducible polynomials to generate S-box 2 and P-box 2 respectively. An S or P boxes will be generated from an input byte. A byte will be converted into a polynomial in $\mathbb{Z}_2$. An inverse of the polynomial modulo irreducible polynomial $m_i(x)$ will go through an affine transform as prescribed in generation process of AES S-box.

3.4 MixColumn Operation

In original AES, a mixed column step is basically operated on four bytes of a state array (Daemen and Rijmen, 2002). Each element of the state matrix is multiplication matrix with the corresponding column of the polynomial matrix. Resulting in four bytes in one column will replaces the original column of the state array. A basic mix column is written in matrical equation as follows:

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

In a decryption process, an inverse of mix column operation can be viewed as follows;

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 19 \\ 19 & 14 & 11 & 13 \\ 13 & 19 & 14 & 11 \\ 11 & 13 & 19 & 14 \end{bmatrix} \begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix}$$

A product of two bytes will be computed as a product of two polynomials in finite field modulo an irreducible polynomial $m_i(x)$. Let us take a simple example:

$$\begin{aligned}
167_{10}\cdot 245_{10} \pmod{299_{10}} &= \texttt{A7}\cdot\texttt{F5} \pmod{\texttt{12B}}\\
&= 10100111\cdot 11110101 \pmod{100101011}\\
&= 112213253321211\\
&= 110011011101011 \pmod{2}\\
&\oplus \underline{100101011}\\
&= 10110000101011\\
&\oplus \underline{100101011}\\
&= 100101001011\\
&\oplus \underline{100101011}\\
&= 10011 \pmod{100101011}
\end{aligned}$$

From this simple example, it can be deduced that, this byte multiplication can be efficiently done in binary mode. An addition will take an exclusive-or of two bytes. Consequently, a mix column operation on a state array can be viewed as a product of two matrices between a mix column matrix and the state array.

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3}\\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3}\\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3}\\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 2\\ 1 & 1 & 3 & 1\\ 1 & 1 & 3 & 1\\ 2 & 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3}\\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3}\\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3}\\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

It should be noted here that a product mix column matrix and its inverse will produce an identity matrix.

$$\begin{bmatrix} 14 & 11 & 19 & 14\\ 13 & 19 & 11 & 13\\ 13 & 19 & 11 & 13\\ 14 & 11 & 19 & 14 \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 & 2\\ 1 & 1 & 3 & 1\\ 1 & 1 & 3 & 1\\ 2 & 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 \end{bmatrix}$$

A mix column is known to be the primary source of diffusion here. In this mega cipher, a larger setting is presented as a mix column operation as shown in Figure 3. A mix column matrix M has been chosen as a diagonal matrix.

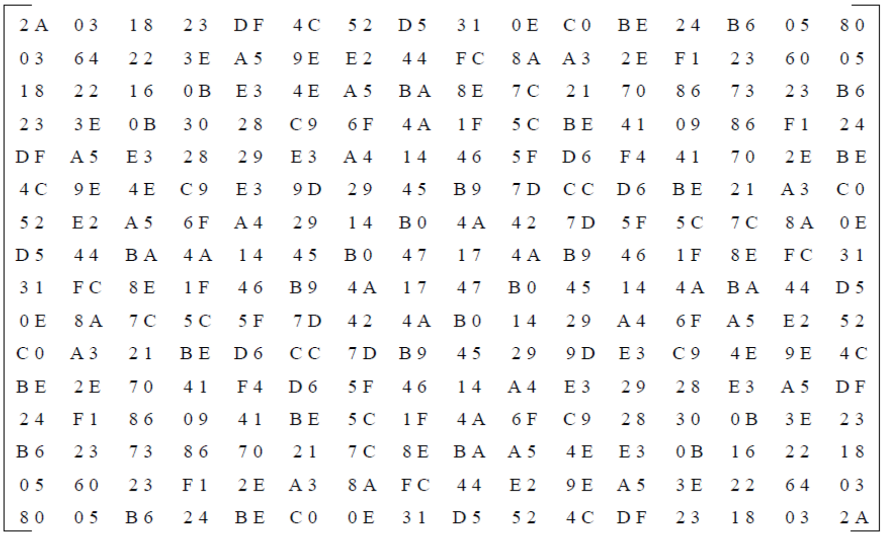
$$\begin{bmatrix} \dot{s}_0 \\ \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \\ \dot{s}_4 \\ \dot{s}_5 \\ \dot{s}_6 \\ \dot{s}_7 \\ \dot{s}_8 \\ \dot{s}_9 \\ \dot{s}_{10} \\ \dot{s}_{11} \\ \dot{s}_{12} \\ \dot{s}_{13} \\ \dot{s}_{14} \\ \dot{s}_{15} \end{bmatrix} = \begin{bmatrix} 3&2&1&&&&&&&&&&&\cdots&1&1 \\ 2&3&2&1&&&&&&&&&&\cdots&1&1 \\ 1&2&3&2&1&&&&&&&&&\ddots&\vdots&\vdots \\ &1&2&3&2&1&&&&&&&&&& \\ &&1&2&3&2&1&&&&&&&&& \\ &&&1&2&3&2&1&&&&&&&& \\ &&&&1&2&3&2&1&&&&&&& \\ &&&&&1&2&3&2&1&&&&&& \\ &&&&&&1&2&3&2&1&&&&& \\ &&&&&&&1&2&3&2&1&&&& \\ &&&&&&&&1&2&3&2&1&&& \\ &&&&&&&&&1&2&3&2&1&& \\ &&&&&&&&&&1&2&3&2&1& \\ \vdots&\vdots&\ddots&&&&&&&&&1&2&3&2&1 \\ 1&1&\cdots&&&&&&&&&&1&2&3&2 \\ 1&1&\cdots&&&&&&&&&&&1&2&3 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \\ s_{15} \end{bmatrix}$$

Figure 3: A Mix Column operation on 16×16 byte array

In this cipher, each mix column operation will have different irreducible polynomials namely, $m_1(x)$, $m_2(x)$, $m_6(x)$ and $m_7(x)$ respectively. Using the same mix column matrix M, there will be 4 different inverse mix column matrices as tabulated in Tables 7-10. An encryption process is typically set simpler in a more efficient mode than decryption. Here, a matrix multiplication on mix column matrix M has been preferably chosen during an encryption operation while an inversion will do the decryption. Both processes will require modular operation. In order to design an efficient cipher, a computationally friendly operation shall be preset in mind.

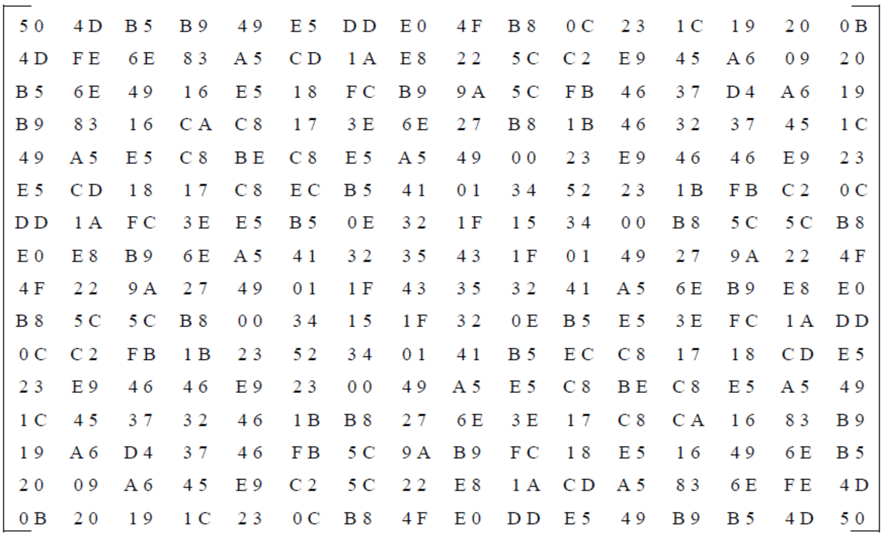
2B	B0	76	69	5C	65	7E	55	7C	80	01	03	67	C4	62	05
B0	23	97	3A	A1	07	58	94	43	15	A6	CB	D7	4B	BF	62
76	97	5F	31	B2	C7	FB	A6	15	73	84	A1	89	9C	4B	C4
69	3A	31	89	B4	25	1F	A2	4E	A2	1F	25	B4	89	D7	67
5C	A1	B2	B4	A3	59	04	71	93	5A	36	B7	25	A1	CB	03
65	07	C7	25	59	0A	F2	0E	5E	C2	7A	36	1F	84	A6	01
7E	58	FB	1F	04	F2	4D	6E	EC	33	C2	5A	A2	73	15	80
55	94	A6	A2	71	0E	6E	51	FD	EC	5E	93	4E	15	43	7C
7C	43	15	4E	93	5E	EC	FD	51	6E	0E	71	A2	A6	94	55
80	15	73	A2	5A	C2	33	EC	6E	4D	F2	04	1F	FB	58	7E
01	A6	84	1F	36	7A	C2	5E	0E	F2	0A	59	25	C7	07	65
03	CB	A1	25	B7	36	5A	93	71	04	59	A3	B4	B2	A1	5C
67	D7	89	B4	25	1F	A2	4E	A2	1F	25	B4	89	31	3A	69
C4	4B	9C	89	A1	84	73	15	A6	FB	C7	B2	31	5F	97	76
62	BF	4B	D7	CB	A6	15	43	94	58	07	A1	3A	97	23	B0
05	62	C4	67	03	01	80	7C	55	7E	65	5C	69	76	B0	2B

Table 7: An inverse mix column matrix $M^{-1} \pmod{m_1(x)}$.



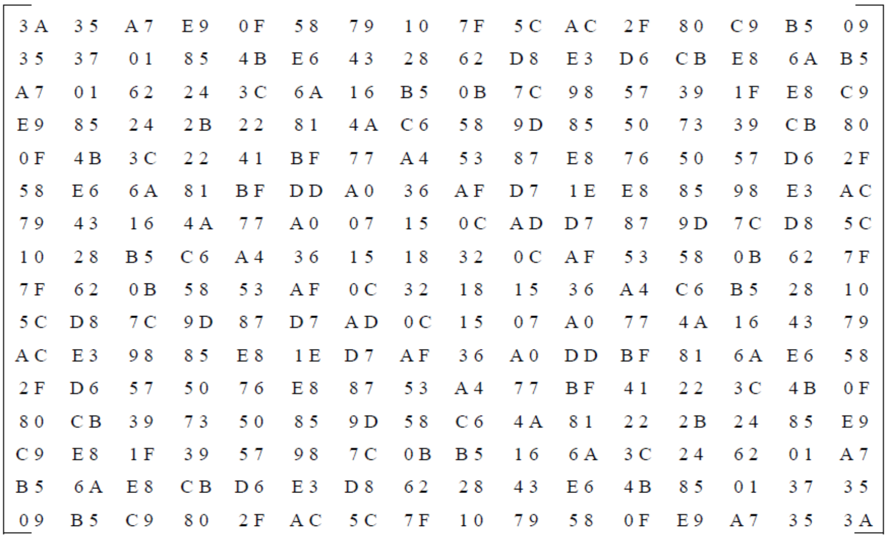
2A	03	18	23	DF	4C	52	D5	31	0E	C0	BE	24	B6	05	80
03	64	22	3E	A5	9E	E2	44	FC	8A	A3	2E	F1	23	60	05
18	22	16	0B	E3	4E	A5	BA	8E	7C	21	70	86	73	23	B6
23	3E	0B	30	28	C9	6F	4A	1F	5C	BE	41	09	86	F1	24
DF	A5	E3	28	29	E3	A4	14	46	5F	D6	F4	41	70	2E	BE
4C	9E	4E	C9	E3	9D	29	45	B9	7D	CC	D6	BE	21	A3	C0
52	E2	A5	6F	A4	29	14	B0	4A	42	7D	5F	5C	7C	8A	0E
D5	44	BA	4A	14	45	B0	47	17	4A	B9	46	1F	8E	FC	31
31	FC	8E	1F	46	B9	4A	17	47	B0	45	14	4A	BA	44	D5
0E	8A	7C	5C	5F	7D	42	4A	B0	14	29	A4	6F	A5	E2	52
C0	A3	21	BE	D6	CC	7D	B9	45	29	9D	E3	C9	4E	9E	4C
BE	2E	70	41	F4	D6	5F	46	14	A4	E3	29	28	E3	A5	DF
24	F1	86	09	41	BE	5C	1F	4A	6F	C9	28	30	0B	3E	23
B6	23	73	86	70	21	7C	8E	BA	A5	4E	E3	0B	16	22	18
05	60	23	F1	2E	A3	8A	FC	44	E2	9E	A5	3E	22	64	03
80	05	B6	24	BE	C0	0E	31	D5	52	4C	DF	23	18	03	2A

Table 8: An inverse mix column matrix $M^{-1} \pmod{m_2(x)}$.



50	4D	B5	B9	49	E5	DD	E0	4F	B8	0C	23	1C	19	20	0B
4D	FE	6E	83	A5	CD	1A	E8	22	5C	C2	E9	45	A6	09	20
B5	6E	49	16	E5	18	FC	B9	9A	5C	FB	46	37	D4	A6	19
B9	83	16	CA	C8	17	3E	6E	27	B8	1B	46	32	37	45	1C
49	A5	E5	C8	BE	C8	E5	A5	49	00	23	E9	46	46	E9	23
E5	CD	18	17	C8	EC	B5	41	01	34	52	23	1B	FB	C2	0C
DD	1A	FC	3E	E5	B5	0E	32	1F	15	34	00	B8	5C	5C	B8
E0	E8	B9	6E	A5	41	32	35	43	1F	01	49	27	9A	22	4F
4F	22	9A	27	49	01	1F	43	35	32	41	A5	6E	B9	E8	E0
B8	5C	5C	B8	00	34	15	1F	32	0E	B5	E5	3E	FC	1A	DD
0C	C2	FB	1B	23	52	34	01	41	B5	EC	C8	17	18	CD	E5
23	E9	46	46	E9	23	00	49	A5	E5	C8	BE	C8	E5	A5	49
1C	45	37	32	46	1B	B8	27	6E	3E	17	C8	CA	16	83	B9
19	A6	D4	37	46	FB	5C	9A	B9	FC	18	E5	16	49	6E	B5
20	09	A6	45	E9	C2	5C	22	E8	1A	CD	A5	83	6E	FE	4D
0B	20	19	1C	23	0C	B8	4F	E0	DD	E5	49	B9	B5	4D	50

Table 9: An inverse mix column matrix $M^{-1} \pmod{m_6(x)}$.



$$\begin{bmatrix}
3A & 35 & A7 & E9 & 0F & 58 & 79 & 10 & 7F & 5C & AC & 2F & 80 & C9 & B5 & 09 \\
35 & 37 & 01 & 85 & 4B & E6 & 43 & 28 & 62 & D8 & E3 & D6 & CB & E8 & 6A & B5 \\
A7 & 01 & 62 & 24 & 3C & 6A & 16 & B5 & 0B & 7C & 98 & 57 & 39 & 1F & E8 & C9 \\
E9 & 85 & 24 & 2B & 22 & 81 & 4A & C6 & 58 & 9D & 85 & 50 & 73 & 39 & CB & 80 \\
0F & 4B & 3C & 22 & 41 & BF & 77 & A4 & 53 & 87 & E8 & 76 & 50 & 57 & D6 & 2F \\
58 & E6 & 6A & 81 & BF & DD & A0 & 36 & AF & D7 & 1E & E8 & 85 & 98 & E3 & AC \\
79 & 43 & 16 & 4A & 77 & A0 & 07 & 15 & 0C & AD & D7 & 87 & 9D & 7C & D8 & 5C \\
10 & 28 & B5 & C6 & A4 & 36 & 15 & 18 & 32 & 0C & AF & 53 & 58 & 0B & 62 & 7F \\
7F & 62 & 0B & 58 & 53 & AF & 0C & 32 & 18 & 15 & 36 & A4 & C6 & B5 & 28 & 10 \\
5C & D8 & 7C & 9D & 87 & D7 & AD & 0C & 15 & 07 & A0 & 77 & 4A & 16 & 43 & 79 \\
AC & E3 & 98 & 85 & E8 & 1E & D7 & AF & 36 & A0 & DD & BF & 81 & 6A & E6 & 58 \\
2F & D6 & 57 & 50 & 76 & E8 & 87 & 53 & A4 & 77 & BF & 41 & 22 & 3C & 4B & 0F \\
80 & CB & 39 & 73 & 50 & 85 & 9D & 58 & C6 & 4A & 81 & 22 & 2B & 24 & 85 & E9 \\
C9 & E8 & 1F & 39 & 57 & 98 & 7C & 0B & B5 & 16 & 6A & 3C & 24 & 62 & 01 & A7 \\
B5 & 6A & E8 & CB & D6 & E3 & D8 & 62 & 28 & 43 & E6 & 4B & 85 & 01 & 37 & 35 \\
09 & B5 & C9 & 80 & 2F & AC & 5C & 7F & 10 & 79 & 58 & 0F & E9 & A7 & 35 & 3A
\end{bmatrix}$$

Table 10: An inverse mix column matrix $M^{-1} \pmod{m_7(x)}$.

This large cipher has been designed for common input files ranging from 2048-bit which is 256 bytes onwards. Since a standard hash function will be used to generate the round keys, a variable length of session key is also feasible in this cipher. It is also practical to use password as a symmetric key. A minimum of 20 character password is recommended for this cipher. In order to achieve a minimum 120-bit strength 20 alphanumeric characters is currently sufficient to overcome a full brute-force attack.

In AES, all byte polynomial operations are done in a fixed finite field modulo an irreducible polynomial $m_0(x) = x^8 + x^4 + x^3 + x + 1$. In order to compensate a large plaintext block, 4 finite fields have been introduced into this mega cipher modulo 4 irreducible polynomials, specifically, $m_1(x)$, $m_2(x)$, $m_6(x)$ and $m_7(x)$ as given in Table 1.

4 A MEGA BLOCK CIPHER

A megabit block cipher is called for here in order to cater for practical needs. Currently, an input file runs in kilobytes. It is apparent a more practical cipher is much needed in handling daily task of protecting an important document from a user stand point of view without having to go through technical knowledge of encryption. As the current block cipher standard, AES has been well-studied cryptographic construction from which parts of AES are used in many cryptographic designs. A suitable design is sought here by simplifying the operation, increasing state array size and decreasing the number of rounds.

4.1 Round Key Generation

In this new proposal of a large symmetric cipher, Round Key 0, Round Key 1 and Round Key 2 may be pre-generated prior to an encryption or decryption process. Initially, a session key will go through hashing processes via SHA256 eight times to accumulate 2048-bit Round Key 0. First, the last 256-bit Round Key 0 will go through another SHA256 8 times in order to accumulate 2048-bit Round Key 1. Second, the last 256-bit Round Key 1 will go through SHA256 another 8 times in order to accumulate 2048-bit Round Key 2 as shown on the right hand side (RHS) of Figure 4. Each round key will be reshaped into 16 by 16 state byte array.

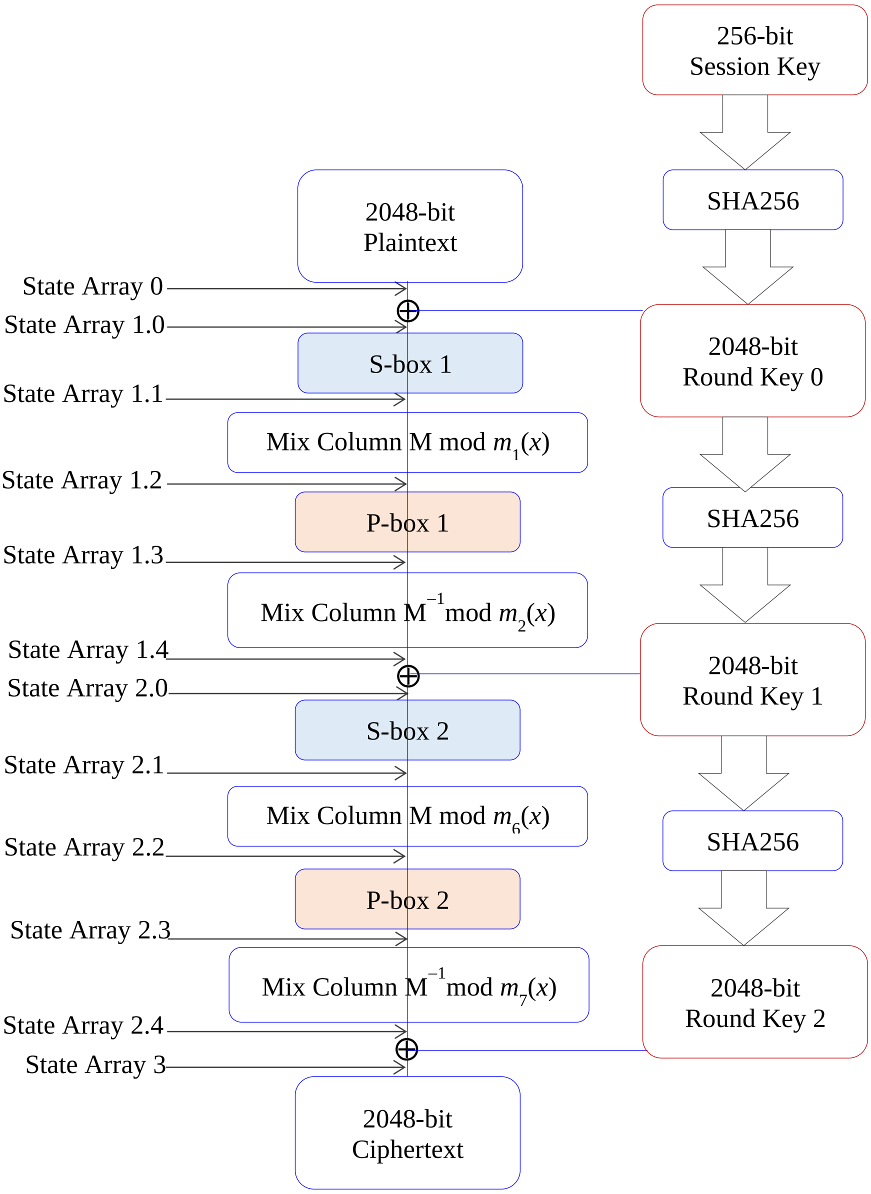


Figure 4: An overview structure of a megabit cipher.

4.2 Encryption Process

Initially, in Round 0, a 2048-bit plaintext will be reshaped into 16 by 16 state array of bytes. The state array will be exclusive-ored with Round Key 0. In Round 1, the state array will first go through S-box 1 and P-box 1. As depicted on the left hand side(LHS) of Figure 4, an incoming 16 by 16 state array will then be exclusive-ored with Round Key 1. Second, the state array will go through S-box 2 and P-box 2 in Round 2. The incoming 16 by 16 state array will then be exclusive-ored with Round Key 2. Nevertheless, a mix column transformation will be injected into the encryption process immediately right after each S or P-box. A combination of S-box and P-box is not expected to provide sufficient full confusion and diffusion effects to this large mega cipher. There will be 4 mix column matrices here, namely, $M \pmod{m_1(x)}$, $M^{-1} \pmod{m_2(x)}$, $M \pmod{m_6(x)}$ and $M^{-1} \pmod{m_7(x)}$.

4.3 Decryption Process

Following an encryption process, Round Key 0, Round Key 1 and Round Key 2 may be pre-generated prior to a decryption process. There are also inverse boxes S^{-1}box 1, S^{-1}box 2, P^{-1}box 1 and P^{-1}box 2 being prescribed for a decryption process. In Round 0, a 2048-bit ciphertext will be reshaped into 16 by 16 state array of bytes. An incoming 16 by 16 state array will be initially exclusive-ored with Round Key 2. In Round 1, the state array will go through P^{-1}box 2 and S^{-1}box 2. The outgoing 16 by 16 state array will be exclusive-ored with Round Key 1. In round 2, the state array will go through P^{-1}box 1 and S^{-1}box 1 consecutively. Lastly, the state array will be exclusive-ored with Round Key 0 in order to produce an original 16 by 16 state array of plaintext bytes.

A mix column transformation will be invoked into the decryption process immediately right after each S or P-box. There will be 4 mix column matrices here, namely, $M \pmod{m_7(x)}$, $M^{-1} \pmod{m_6(x)}$, $M \pmod{m_2(x)}$ and $M^{-1} \pmod{m_1(x)}$.

4.4 Sample Input and Output

In light of current corona pandemic during which this paper has been written on, a sample plaintext has been chosen from a poem describing a future corona pandemic about more than 800 years earlier.

```
1.  By the turn of the year 20 century a thousand and a thousand; 2.
The world is attacked by corona as a result of wicked human, 14.  They
disseminate information ~they study it to no avail, 15.  The good and
the bad are afraid ~they bury their dead, 18.~
```

The session key for this sample is as follows:

```
Abu Ali Al-Dabizi, Baghdad 1170.
```

Nevertheless, the real message is supposed to be the verse from line

```
18.  The kings of the earth are in despair ...  spending too much money
fighting it.
```

Every task description during the operational process on each state array has been prescribed in Table 11. They have been pointed out on their locations in the overall design on this mega cipher in Figure 4.

State Array	Task Description	First Input Matrix	Second Input Matrix
0	Form initial column wise State Array	1D plaintext	
1.0	An exclusive-or against Round Key	State Array 0	Round Key 0
1.1	Byte Substitution operation	State Array 1.0	S-box 1
1.2	Mix Column Matrix Multiplication	$M \pmod{m_1(x)}$	State Array 1.1
1.3	Byte Permutation operation State	Array 1.2	P-box 1
1.4	Mix Column Matrix Multiplication	$M^{-1} \pmod{m_2(x)}$	State Array 1.3
2.0	An exclusive-or against Round Key	State Array 1.4	Round Key 1
2.1	Byte Substitution operation	State Array 2.0	S-box 2
2.2	Mix Column Matrix Multiplication	$M^-1 \pmod{m_3(x)}$	State Array 2.1
2.3	Byte Permutation operation State	Array 2.2	P-box 2
2.4	Mix Column Matrix Multiplication	$M \pmod{m_4(x)}$	State Array 2.3
3.0	An exclusive-or against Round Key	State Array 2.4	Round Key 2

Table 11: A task description on every State Array.

On one hand, step by step state array matrices during an encryption process of this Mega Cipher are given in Appendix 1. Next to the state array, another matrix on the next operation is also given. On another hand, step by step state array matrices during a decryption process of this Mega Cipher are given in Appendix 2.

At each stage, a 2048-bit plaintext will be presented as 16 by 16 state array of bytes. Every operation in this mega cipher has been also compactly represented as a 16 by 16 matrix of bytes. They are being tabulated in Appendices during an encryption and decryption processes.

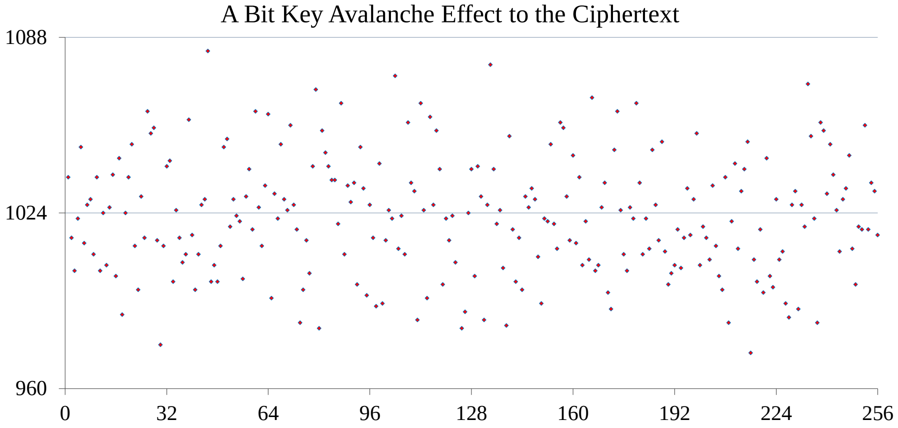


Figure 5: An avalanche effect of a bit change on the 256-bit session key.

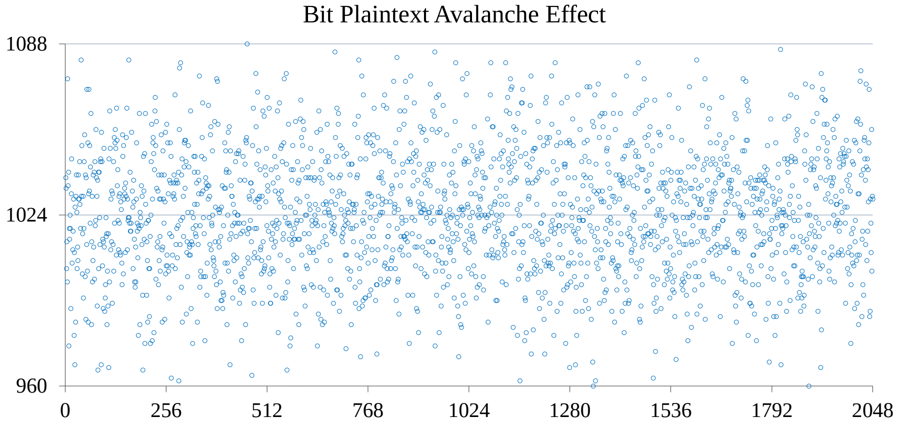


Figure 6: An avalanche effect from a bit change in the 2048-bit plaintext to the ciphertext.

4.5 An Avalanche Effect to the Ciphertext

A preliminary avalanche effect test on a bit change in the 256-bit session key has produced an average score of 1022.63 bits change on the ciphertext. The resulting bit changes in the ciphertext have been depicted in Figure 5. At the same time, an avalanche effect test on a bit change in the 2048-bit sample plaintext has produced an average score of 1024.55 bits change on the sample ciphertext. The resulting bit changes in the ciphertext have been depicted in Figure 6. This mega cipher has managed to achieve a strict avalanche criterion in two rounds (Castro et al., 2005).

5 CONCLUSION

In its original proposal, AES can have a variation of key lengths, namely, 128 bits, 192 bits or 256 bits on the same plaintext/ciphertext block size of 128 bits. In this mega cipher, a larger 2056-bit plaintext/ciphertext has been introduced using a variable key length. A larger block size will certainly open a larger room for possible attack on this mega cipher. A strong and in depth attack and cryptanalysis are very much welcomed here. From a practical design, this mega cipher has a potential to be popular cipher to non-technical users in various application around the globe.

REFERENCES

Bertoni, G., Breveglieri, L., Fragneto, P., Macchetti, M., and Marchesin, S. (2002). Efficient software implementation of aes on 32-bit platforms. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 159–171. Springer.

Castro, J. C. H., Sierra, J. M., Seznec, A., Izquierdo, A., and Ribagorda, A. (2005). The strict avalanche criterion randomness test. *Mathematics and Computers in Simulation*, 68(1):1–7.

Daemen, J. and Rijmen, V. (2002). *The design of Rijndael*, volume 2. Springer.

Gladman, B. (2002). Implementations of aes (rijndael) in c/c++ and assembler. *http://fp. gladman. plus. com/cryptography_technology/rijndael*.

Rijmen, V. and Daemen, J. (2001). Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pages 19–22.

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 0: Plaintext

31	66	6E	20	20	20	6F	6F	20	69	6E	74	35	74	69	68
2E	20	74	61	32	61	6E	66	31	6E	20	20	2E	68	64	65
20	74	75	6E	2E	74	61	20	34	61	7E	74	20	65	20	69
42	68	72	64	20	74	20	77	2E	74	20	6F	54	20	7E	72
79	65	79	20	54	61	61	69	20	65	74	20	68	62	20	20
20	20	20	61	68	63	73	63	54	20	68	6E	65	61	74	64
74	79	61	20	65	6B	20	6B	68	69	65	6F	20	64	68	65
68	65	20	74	20	65	61	65	65	6E	79	20	67	20	65	61
65	61	74	68	77	64	20	64	79	66	20	61	6F	61	79	64
20	72	68	6F	6F	20	72	20	20	6F	73	76	6F	72	20	2C
74	20	6F	75	72	62	65	68	64	72	74	61	64	65	62	20
75	32	75	73	6C	79	73	75	69	6D	75	69	20	20	75	31
72	30	73	61	64	20	75	6D	73	61	64	6C	61	61	72	38
6E	20	61	6E	20	63	6C	61	73	74	79	2C	6E	66	79	2E
20	63	6E	64	69	6F	74	6E	65	69	20	20	64	72	20	20
6F	65	64	2C	73	72	20	2C	6D	6F	69	31	20	61	74	7E

Round Key 0

CA	51	29	61	E4	4B	3C	C0	D4	F9	5C	2F	8D	2B	3F	10
CA	6C	19	D9	58	71	1C	7F	EA	6D	63	11	7C	F5	B6	B1
1C	3E	C9	05	00	3F	D0	D4	1D	5F	A8	58	FA	4F	3A	14
1A	B8	15	82	6C	E7	46	D1	0C	92	8C	EE	EF	10	F8	3B
5A	A8	1F	97	DD	43	C6	36	47	33	B2	0C	68	E0	6E	BF
16	38	26	BD	94	65	63	8A	81	CA	04	0C	17	9D	82	F3
8A	82	E8	67	FE	6E	A1	33	CE	3C	09	1D	66	F3	77	18
F2	47	AF	6A	4B	49	D9	B4	2B	FF	CE	49	74	43	8C	CB
F5	B2	82	1D	19	D3	EC	04	C2	98	D8	DA	9C	40	CA	4B
68	A8	A1	65	EF	CE	F3	D9	93	A0	3E	D0	92	EF	58	CF
E5	17	E6	8E	B4	8E	DB	38	0B	DD	60	A0	E4	78	51	77
1E	6B	0A	3F	01	61	21	03	BA	47	64	25	2D	51	55	98
FD	FE	A4	F4	0E	C3	70	20	CE	D3	DD	20	51	43	E4	51
E8	BC	DE	5F	B9	12	3D	72	83	8B	12	3C	DA	77	0E	83
6E	4F	27	88	D2	95	FB	DC	D5	26	5E	2F	6B	8D	D4	91
40	DA	43	D5	BD	40	B8	C9	CC	04	89	B6	1C	D9	58	18

State Array 1.0

FB	37	47	41	C4	6B	53	AF	F4	90	32	5B	B8	5F	56	78
E4	4C	6D	B8	6A	10	72	19	DB	03	43	31	52	9D	D2	D4
3C	4A	BC	6B	2E	4B	B1	F4	29	3E	D6	2C	DA	2A	1A	7D
58	D0	67	E6	4C	93	66	A6	22	E6	AC	81	BB	30	86	49
23	CD	66	B7	89	22	A7	5F	67	56	C6	2C	00	82	4E	9F
36	18	06	DC	FC	06	10	E9	D5	EA	6C	62	72	FC	F6	97
FE	FB	89	47	9B	05	81	58	A6	55	6C	72	46	97	1F	7D
9A	22	8F	1E	6B	2C	B8	D1	4E	91	B7	69	13	63	E9	AA
90	D3	F6	75	6E	B7	CC	60	BB	FE	F8	BB	F3	21	B3	2F
48	DA	C9	0A	80	EE	81	F9	B3	CF	4D	A6	FD	9D	78	E3
91	37	89	FB	C6	EC	BE	50	6F	AF	14	C1	80	1D	33	57
6B	59	7F	4C	6D	18	52	76	D3	2A	11	4C	0D	71	20	A9
8F	CE	D7	95	6A	E3	05	4D	BD	B2	B9	4C	30	22	96	69
86	9C	BF	31	99	71	51	13	F0	FF	6B	10	B4	11	77	AD
4E	2C	49	EC	BB	FA	8F	B2	B0	4F	7E	0F	0F	FF	F4	B1
2F	BF	27	F9	CE	32	98	E5	A1	6B	E0	87	3C	B8	2C	66

S-box Matrix S_1 mod $m_1(x)$

63	7C	7E	8A	7F	27	97	73	FF	8F	D3	36	8B	91	6B	A0
2D	DD	87	C1	3B	B2	5B	2E	17	55	1A	DB	67	50	10	E5
D6	02	AE	30	83	D7	32	8D	4F	16	19	71	ED	F4	57	EA
59	06	78	09	4D	E1	3F	D4	F3	58	68	93	48	25	20	2C
2B	45	41	D8	85	5E	CA	BD	13	49	AB	69	CB	33	86	1C
75	08	D9	BF	CC	BA	6A	4A	24	F1	A8	77	79	40	35	E2
EC	96	D1	5F	EE	AD	C4	54	74	C6	B0	3D	DF	A7	2A	F0
B9	07	6C	21	E6	A2	1B	F2	64	F6	D2	53	C2	92	56	5C
47	89	70	4C	E0	84	BE	2F	82	15	FD	EF	B7	8C	0C	43
C9	9F	E4	A3	95	5D	66	CE	37	0F	4B	05	03	1E	DC	C0
FA	28	44	CF	3E	88	0D	FE	26	6D	1D	80	E7	8E	65	C5
52	12	B8	C3	14	0A	FB	3C	6E	46	60	00	DA	B5	31	D0
A4	5A	0B	9D	3A	F5	7D	B4	A5	29	04	EB	22	81	F8	94
7A	AA	23	BC	18	B6	DE	AC	AF	9E	01	99	C7	9A	38	1F
9C	E3	51	7B	76	62	42	61	A1	B1	11	0E	CD	6F	39	E8
72	F7	A9	A6	BB	34	E9	4E	B3	98	9B	90	F9	D5	FC	C8

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 1.1

90	D4	BD	45	3A	3D	BF	C5	BB	C9	78	77	6E	E2	6A	64
76	CB	A7	6E	B0	2D	6C	55	99	8A	D8	06	D9	1E	23	18
48	AB	DA	3D	57	69	12	BB	16	20	DE	ED	01	19	1A	92
24	7A	54	42	CB	A3	C4	0D	AE	42	E7	89	00	59	BE	49
30	81	C4	3C	15	AE	FE	E2	54	6A	7D	ED	63	70	86	C0
3F	17	97	C7	F9	97	2D	B1	B6	11	DF	D1	6C	F9	E9	CE
FC	90	15	BD	05	27	89	24	0D	BA	DF	6C	CA	CE	E5	92
4B	AE	43	10	3D	ED	6E	AA	86	9F	3C	C6	C1	5F	B1	1D
C9	BC	E9	A2	2A	3C	22	EC	00	FC	B3	00	A6	02	C3	EA
13	01	29	D3	47	39	89	98	C3	94	33	0D	D5	1E	64	7B
9F	D4	15	90	7D	CD	31	75	F0	C5	3B	5A	47	50	09	4A
3D	F1	5C	CB	A7	17	D9	1B	BC	19	DD	CB	91	07	D6	6D
43	F8	AC	5D	B0	7B	27	33	B5	B8	46	CB	59	AE	66	C6
BE	03	D0	06	0F	07	08	C1	72	C8	3D	2D	14	DD	F2	8E
86	ED	49	CD	00	9B	43	B8	52	1C	56	A0	A0	C8	BB	12
EA	D0	8D	98	F8	78	37	62	28	3D	9C	2F	48	6E	ED	C4

Mix Column Matrix M mod $m_1(x)$

03	02	01	01	01	01	01	01	01	01	01	01	01	01	01	01
02	03	02	01	01	01	01	01	01	01	01	01	01	01	01	01
01	02	03	02	01	01	01	01	01	01	01	01	01	01	01	01
01	01	02	03	02	01	01	01	01	01	01	01	01	01	01	01
01	01	01	02	03	02	01	01	01	01	01	01	01	01	01	01
01	01	01	01	02	03	02	01	01	01	01	01	01	01	01	01
01	01	01	01	01	02	03	02	01	01	01	01	01	01	01	01
01	01	01	01	01	01	02	03	02	01	01	01	01	01	01	01
01	01	01	01	01	01	01	02	03	02	01	01	01	01	01	01
01	01	01	01	01	01	01	01	02	03	02	01	01	01	01	01
01	01	01	01	01	01	01	01	01	02	03	02	01	01	01	01
01	01	01	01	01	01	01	01	01	01	02	03	02	01	01	01
01	01	01	01	01	01	01	01	01	01	01	02	03	02	01	01
01	01	01	01	01	01	01	01	01	01	01	01	02	03	02	01
01	01	01	01	01	01	01	01	01	01	01	01	01	02	03	02
01	01	01	01	01	01	01	01	01	01	01	01	01	01	02	03

State Array 1.2

4A	CD	33	1E	BD	64	C8	97	1C	96	BC	92	2E	F7	DD	EE
74	04	6C	72	CE	CF	2D	E1	04	B5	55	FF	9A	20	BA	0F
BD	BD	01	28	11	02	DE	7C	B4	A9	D9	3D	F0	C1	D4	F2
1B	B2	2A	A1	49	66	95	38	70	C0	24	4F	14	33	B4	6A
96	A6	6D	FA	4E	42	EE	C9	41	BB	BD	6F	C0	11	B2	07
DA	25	FD	2B	DB	DC	EA	E1	6D	E3	57	57	BA	0A	30	4F
94	D3	DD	25	5F	A9	D5	AC	8B	7C	94	97	D1	4C	65	5F
12	3B	09	27	39	B5	23	C5	F1	45	C3	65	AF	FB	4F	BC
8A	B1	E7	27	E8	46	6F	6C	0E	54	53	00	E9	FD	95	5B
07	82	DD	FD	45	23	25	52	57	D2	DA	82	3B	F0	D1	33
BC	80	15	05	C3	AA	BB	8D	8B	87	60	A3	F0	B1	83	A0
EE	85	F8	E7	2B	B6	8A	35	5D	2F	19	53	99	1D	5A	6B
F3	DE	6C	E0	AA	AF	3F	EA	F9	99	88	CA	A4	08	CC	A7
E8	01	2F	B1	D7	6C	95	D6	27	E6	45	91	BA	01	EF	6E
00	97	D5	1E	00	F5	EE	5C	8B	96	6B	1B	3D	7F	10	F4
85	AF	4A	41	E9	1F	82	EE	67	C4	E6	E3	E9	95	7B	9B

P-box Matrix P_1 mod $m_2(x)$

63	7C	AA	ED	C1	E6	24	88	32	3A	A1	3F	86	33	96	64
CB	E5	CF	14	44	45	0B	AC	D7	3C	4B	54	DF	A8	A6	80
37	B9	20	A5	73	BC	D8	5E	F0	1D	70	A9	11	0A	84	2D
7F	A2	8A	65	31	4E	F8	99	7B	D9	C0	09	81	29	92	FA
0F	EB	48	69	C2	41	00	DE	6B	B8	8C	8E	BE	BA	FD	4D
EC	BF	5C	A7	EA	9E	40	CC	1C	CA	91	62	D6	C4	02	78
2B	35	C5	AE	97	21	26	82	4A	F3	F5	36	E8	FE	1E	52
6F	59	3E	3B	B2	03	10	BB	12	2E	46	B6	9B	25	E9	27
55	A0	61	30	B0	98	66	DA	B3	D0	34	58	94	AB	FB	72
67	EF	C8	75	D2	2F	D3	17	8D	D4	C9	CE	2C	E7	74	43
A4	F4	0D	51	FC	A3	01	E2	E1	C3	DB	D1	B4	68	F2	5D
DC	F7	B7	16	1A	39	E3	6C	FF	3D	F6	13	95	50	EE	5A
47	2A	0E	1B	76	9A	85	57	5F	08	42	B5	87	90	93	7D
B1	79	6D	56	28	9F	8F	AF	E0	19	AD	D5	DD	C7	BD	71
23	6A	38	0C	8B	77	4F	7A	CD	7E	15	04	9C	18	49	E4
9D	05	83	53	F1	5B	89	C6	1F	F9	06	22	60	6E	07	4C

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 1.3

EE	BB	30	B5	1B	AF	E6	7B	99	4F	C1	2D	1E	15	6C	96
23	F0	F1	53	72	6B	E7	52	7F	E6	2B	E0	6D	A9	65	67
01	A9	E3	00	C8	FB	D5	BC	D7	33	DE	94	3B	F2	45	23
27	49	1C	F7	53	D3	97	BD	D5	B6	96	27	B5	2F	09	92
EA	42	88	33	CE	CF	C3	F3	6D	10	8B	55	9B	07	66	EE
1D	05	5F	41	FF	8A	B1	EA	00	3B	6B	1F	FD	A0	7C	F9
E9	E7	57	4A	EE	A1	6F	07	B1	FA	97	41	35	2F	95	12
D9	6E	5B	11	D1	FD	AA	F5	4F	01	5C	70	CD	A7	96	1B
0F	14	AC	4A	D4	3F	2E	A4	97	82	2A	00	BD	57	6F	95
08	57	B4	CC	E9	99	DD	5F	46	38	AF	AF	3D	85	DC	6C
B1	BC	B2	AA	BC	28	BA	2B	20	3D	33	FD	E1	45	25	D6
E8	E8	39	0E	F0	CA	65	F8	BB	BD	11	C5	02	EF	C0	25
24	BD	4E	87	0A	DD	EE	01	DD	DA	E3	74	E1	8B	82	6C
54	A3	45	25	D2	91	BA	04	DE	C0	6C	60	EE	BA	C9	9A
27	8B	8D	8A	F4	04	64	F0	D1	4F	DB	A6	DA	1E	5A	82
B4	E9	83	7C	80	94	19	85	95	C4	6A	95	C3	B2	4C	5D

Inverse Mix Column Matrix M^{-1} mod $m_2(x)$

2A	03	18	23	DF	4C	52	D5	31	0E	C0	BE	24	B6	05	80
03	64	22	3E	A5	9E	E2	44	FC	8A	A3	2E	F1	23	60	05
18	22	16	0B	E3	4E	A5	BA	8E	7C	21	70	86	73	23	B6
23	3E	0B	30	28	C9	6F	4A	1F	5C	BE	41	09	86	F1	24
DF	A5	E3	28	29	E3	A4	14	46	5F	D6	F4	41	70	2E	BE
4C	9E	4E	C9	E3	9D	29	45	B9	7D	CC	D6	BE	21	A3	C0
52	E2	A5	6F	A4	29	14	B0	4A	42	7D	5F	5C	7C	8A	0E
D5	44	BA	4A	14	45	B0	47	17	4A	B9	46	1F	8E	FC	31
31	FC	8E	1F	46	B9	4A	17	47	B0	45	14	4A	BA	44	D5
0E	8A	7C	5C	5F	7D	42	4A	B0	14	29	A4	6F	A5	E2	52
C0	A3	21	BE	D6	CC	7D	B9	45	29	9D	E3	C9	4E	9E	4C
BE	2E	70	41	F4	D6	5F	46	14	A4	E3	29	28	E3	A5	DF
24	F1	86	09	41	BE	5C	1F	4A	6F	C9	28	30	0B	3E	23
B6	23	73	86	70	21	7C	8E	BA	A5	4E	E3	0B	16	22	18
05	60	23	F1	2E	A3	8A	FC	44	E2	9E	A5	3E	22	64	03
80	05	B6	24	BE	C0	0E	31	D5	52	4C	DF	23	18	03	2A

State Array 1.4

C3	CE	A5	40	6A	3A	8B	79	2C	08	2C	E8	AC	15	DB	D0
33	F5	77	01	AC	E4	72	97	C3	3F	42	C9	2F	29	05	3D
DE	E5	C4	A8	65	C3	9D	88	07	BF	7C	91	50	B5	4D	C1
86	3C	0F	16	8E	57	C0	E3	08	3B	A0	39	7E	9D	36	B8
19	E2	AE	F3	49	45	B2	B1	0D	F5	FB	C0	4F	7A	DD	B0
25	63	D7	B7	53	0C	4A	49	6C	52	8B	E8	90	F0	31	ED
0A	7E	A0	04	97	53	73	BC	0C	8A	D0	D2	DC	86	64	4B
50	B7	4E	9B	CF	69	EF	B9	42	37	53	29	33	2D	7C	8B
03	96	89	3E	BB	2E	7D	D1	C9	A5	B3	AE	9D	24	38	AA
BC	82	B0	9B	B8	E5	B4	AB	3D	85	A2	4A	96	D1	0F	2E
38	4F	51	4F	10	87	F9	C6	94	41	EA	10	4A	30	00	A0
DD	41	D4	50	25	AA	AE	13	8C	2F	FF	63	9D	3B	6A	73
1D	B4	7F	F2	5A	A0	CC	A9	89	DF	7A	73	4F	7B	BF	FB
3C	2F	D0	56	A6	8C	6B	32	B5	F9	C7	BA	90	2C	B8	2A
27	F7	7D	52	39	8A	FF	AC	99	3B	24	28	18	19	9B	6E
F0	E4	5C	5E	F8	D1	3F	8F	5D	E4	83	8A	EE	C1	1A	CF

Round Key 1

CE	1D	8E	64	9E	7D	FE	63	69	F7	C0	46	1C	74	7F	97
EC	DE	F0	C0	2F	0F	15	40	69	2F	E7	FE	54	86	7E	C4
4C	DA	B5	85	41	13	7F	9C	BB	FB	29	A3	11	5E	84	88
D0	11	86	AE	7E	AA	04	83	66	76	29	F1	E5	D2	7B	F5
28	75	FC	76	F6	50	F6	04	EF	3E	18	0C	65	2F	11	A4
70	A3	94	10	73	17	28	61	07	5D	F9	E5	10	86	DC	93
39	3D	10	51	E0	CC	4A	2C	88	84	1A	82	16	76	3C	D2
0D	FE	1E	9E	3D	95	0C	32	B4	30	4B	DA	4C	6A	06	72
DA	0D	C3	97	50	41	22	69	D4	A0	C7	58	02	9F	A0	87
4D	03	91	1F	B8	57	B6	1F	A3	45	74	4D	93	A0	02	B2
18	91	AC	5C	BA	98	13	AC	8D	02	5E	4D	13	CC	D2	87
7B	90	F8	E6	7E	F5	DB	72	F4	F0	89	84	72	9C	59	16
C1	E2	8A	08	2E	2F	A8	E5	DB	FC	86	76	BF	4C	24	2F
4A	A9	E1	88	95	E3	DE	27	57	1E	13	D1	89	01	96	74
13	6E	27	2B	8E	86	D7	E3	1E	DF	82	DD	F7	88	E5	18
92	87	56	BA	38	9A	B9	8D	3D	54	EB	07	3D	9B	91	96

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 2.0

0D	D3	2B	24	F4	47	75	1A	45	FF	EC	AE	B0	61	A4	47
DF	2B	87	C1	83	EB	67	D7	AA	10	A5	37	7B	AF	7B	F9
92	3F	71	2D	24	D0	E2	14	BC	44	55	32	41	EB	C9	49
56	2D	89	B8	F0	FD	C4	60	6E	4D	89	C8	9B	4F	4D	4D
31	97	52	85	BF	15	44	B5	E2	CB	E3	CC	2A	55	CC	14
55	C0	43	A7	20	1B	62	28	6B	0F	72	0D	80	76	ED	7E
33	43	B0	55	77	9F	39	90	84	0E	CA	50	CA	F0	58	99
5D	49	50	05	F2	FC	E3	8B	F6	07	18	F3	7F	47	7A	F9
D9	9B	4A	A9	EB	6F	5F	B8	1D	05	74	F6	9F	BB	98	2D
F1	81	21	84	00	B2	02	B4	9E	C0	D6	07	05	71	0D	9C
20	DE	FD	13	AA	1F	EA	6A	19	43	B4	5D	59	FC	D2	27
A6	D1	2C	B6	5B	5F	75	61	78	DF	76	E7	EF	A7	33	65
DC	56	F5	FA	74	8F	64	4C	52	23	FC	05	F0	37	9B	D4
76	86	31	DE	33	6F	B5	15	E2	E7	D4	6B	19	2D	2E	5E
34	99	5A	79	B7	0C	28	4F	87	E4	A6	F5	EF	91	7E	76
62	63	0A	E4	C0	4B	86	02	60	B0	68	8D	D3	5A	8B	59

S-box Matrix S_2 mod $m_6(x)$

63	7C	D7	44	02	81	F0	F3	E8	13	12	24	91	74	10	C2
9D	2E	60	28	E0	F4	FB	6E	1A	DA	D3	61	E1	A1	B3	7F
27	45	FE	09	E2	C3	C6	0F	99	CE	A8	26	14	B0	DE	0A
E4	CF	BF	58	3B	A5	62	1C	19	B5	39	46	30	90	56	3C
7A	A9	70	35	AD	7B	6D	32	98	41	33	03	8A	52	55	C9
1E	D6	8E	F8	BD	A7	FA	88	D8	64	B1	6C	86	67	EC	21
A0	50	0E	53	0D	BA	C5	6A	4F	47	00	1D	E3	FD	DC	FC
65	BB	08	E5	4E	57	F1	FF	CA	48	9A	2A	F9	72	F7	84
EF	3E	3D	07	EA	2F	73	93	04	AF	6F	85	5F	76	CB	23
9E	1F	49	D4	4B	CC	68	69	97	17	C0	A3	78	D1	36	A2
DD	D9	82	8D	AE	8C	95	3F	0C	9B	01	4A	94	8B	96	06
BE	16	DB	BC	31	92	DF	C4	AA	89	5A	80	A4	B6	42	C8
B9	F6	C1	25	D5	51	40	77	54	7E	B4	9C	0B	1B	E7	6B
75	05	71	D0	E9	2B	5C	5E	18	D2	2C	7D	87	43	AC	37
5B	5D	34	A6	ED	83	20	4D	F5	8F	79	4C	11	66	2D	E6
B7	59	CD	22	9F	38	C7	B2	15	3A	EB	EE	29	B8	AB	F2

State Array 2.1

74	D0	26	E2	9F	32	57	D3	7B	F2	11	96	BE	50	AE	32
37	26	93	F6	07	4C	6A	5E	01	9D	8C	1C	2A	06	2A	3A
49	3C	BB	B0	E2	75	34	E0	A4	AD	A7	BF	A9	4C	7E	41
FA	B0	AF	AA	B7	B8	D5	A0	DC	52	AF	54	A3	C9	52	52
CF	69	8E	2F	C8	F4	AD	92	34	9C	A6	0B	A8	A7	0B	E0
A7	B9	35	3F	27	61	0E	99	1D	C2	08	74	EF	F1	66	F7
58	35	BE	A7	FF	A2	B5	9E	EA	10	B4	1E	B4	B7	D8	17
67	41	1E	81	CD	29	A6	85	C7	F3	1A	22	84	32	9A	3A
D2	A3	33	9B	4C	FC	21	AA	A1	81	4E	C7	A2	80	97	B0
59	3E	45	EA	63	DB	D7	31	36	B9	5C	F3	81	BB	74	78
27	AC	B8	28	01	7F	79	00	DA	35	31	67	64	29	71	0F
95	05	14	DF	6C	21	57	50	CA	37	F1	4D	E6	3F	58	BA
87	FA	38	EB	4E	23	0D	8A	8E	09	29	81	B7	1C	A3	E9
F1	73	CF	AC	58	FC	92	F4	34	4D	E9	1D	DA	B0	DE	EC
3B	17	B1	48	C4	91	99	C9	93	ED	95	38	E6	1F	F7	F1
0E	53	12	ED	B9	03	73	D7	A0	BE	4F	76	D0	B1	85	64

Inverse Mix Column Matrix M^{-1} mod $m_6(x)$

50	4D	B5	B9	49	E5	DD	E0	4F	B8	0C	23	1C	19	20	0B
4D	FE	6E	83	A5	CD	1A	E8	22	5C	C2	E9	45	A6	09	20
B5	6E	49	16	E5	18	FC	B9	9A	5C	FB	46	37	D4	A6	19
B9	83	16	CA	C8	17	3E	6E	27	B8	1B	46	32	37	45	1C
49	A5	E5	C8	BE	C8	E5	A5	49	00	23	E9	46	46	E9	23
E5	CD	18	17	C8	EC	B5	41	01	34	52	23	1B	FB	C2	0C
DD	1A	FC	3E	E5	B5	0E	32	1F	15	34	00	B8	5C	5C	B8
E0	E8	B9	6E	A5	41	32	35	43	1F	01	49	27	9A	22	4F
4F	22	9A	27	49	01	1F	43	35	32	41	A5	6E	B9	E8	E0
B8	5C	5C	B8	00	34	15	1F	32	0E	B5	E5	3E	FC	1A	DD
0C	C2	FB	1B	23	52	34	01	41	B5	EC	C8	17	18	CD	E5
23	E9	46	46	E9	23	00	49	A5	E5	C8	BE	C8	E5	A5	49
1C	45	37	32	46	1B	B8	27	6E	3E	17	C8	CA	16	83	B9
19	A6	D4	37	46	FB	5C	9A	B9	FC	18	E5	16	49	6E	B5
20	09	A6	45	E9	C2	5C	22	E8	1A	CD	A5	83	6E	FE	4D
0B	20	19	1C	23	0C	B8	4F	E0	DD	E5	49	B9	B5	4D	50

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 2.2

C5	0D	6D	C6	83	57	ED	B3	7D	25	31	E1	E4	FE	23	B0
E4	14	B8	55	D2	05	37	B4	9F	3A	2D	E2	6A	CF	C7	2C
9F	E4	20	7D	47	A5	5D	C7	C9	91	7F	C6	B7	9C	20	CD
C6	EA	BC	5D	12	20	40	49	2C	ED	A7	9C	6F	48	F3	C6
03	6A	C0	43	D2	3D	63	4F	55	FA	DC	8D	24	41	92	24
CF	D3	AA	B5	93	2B	D5	6A	07	67	51	FC	36	00	89	FD
1B	CD	F3	BE	C9	E0	F2	53	05	AC	07	27	5C	45	60	3D
1A	A5	8E	32	AC	84	17	E4	06	4C	3B	E2	8A	40	1E	85
64	66	E1	D3	1F	CF	87	0E	1D	66	9E	AB	D5	C1	9D	8E
E3	D2	CD	56	66	2B	FF	10	2E	25	E9	43	CA	54	1A	42
1E	03	B8	DF	9E	C5	6D	26	DB	65	18	17	9E	9B	83	38
A4	E6	99	EA	69	48	93	B2	AB	22	EA	EB	BA	4B	E9	34
5A	57	B6	A9	10	DB	0E	ED	0E	A7	46	B0	A8	16	9A	94
FB	39	B9	F2	16	0B	91	E4	84	0C	34	DB	1E	B1	FB	69
E9	BB	70	5F	D8	4D	93	9F	BB	C6	84	07	F3	F3	5C	DD
12	B4	DB	2F	9F	C2	09	6E	C7	81	02	E7	E7	4D	E0	F3

P-box Matrix P_2 mod m_7(x)

63	7C	18	31	2A	0A	4A	FA	C7	EB	23	AD	03	3A	5B	BB
C5	D4	D3	E8	43	50	04	54	A7	1D	CF	1B	8B	7E	FB	F9
C4	EA	4C	85	3B	86	52	97	87	FD	0E	B6	D0	4B	F8	1C
01	F2	5C	F1	C1	1A	AB	6B	17	D6	19	14	DB	9F	DA	8D
44	C6	53	A1	F4	9B	E4	37	4F	EE	65	57	0F	4E	ED	71
E5	21	2C	B4	D5	B5	89	C9	BA	3C	83	69	5A	CD	DC	EC
A6	26	5F	BC	FC	99	DE	12	32	68	2B	60	F3	EF	67	98
59	11	4D	DD	AA	29	D8	84	3F	10	E9	B1	BF	77	E0	82
F0	C3	45	CE	8F	90	F6	6F	A8	06	1F	15	A0	2D	BD	AF
75	B8	A5	B2	94	09	79	A3	55	3E	F5	80	24	E3	6A	02
20	51	42	07	30	8E	88	C2	CC	B3	08	96	16	61	36	CA
7B	6D	38	22	13	FF	66	40	0B	B7	C0	3D	48	62	A4	D9
81	7F	35	00	7D	7A	8C	9C	AC	F7	1E	6E	49	A2	2F	6C
CB	92	E6	28	47	39	E2	78	DF	27	25	E7	95	D7	9E	34
8A	41	AE	70	74	33	C8	5E	73	91	46	A9	BE	9A	64	E1
B9	58	2E	5D	D2	D1	FE	72	0D	05	9D	0C	56	B0	93	76

State Array 2.3

A9	C6	42	E4	37	81	66	DF	18	2B	57	AB	E7	C7	7F	24
4C	A5	53	69	9C	AB	9E	2C	6D	A7	20	E2	CD	3A	46	9E
1E	D3	EA	31	CA	34	CD	0C	F2	84	83	07	AA	C1	DB	9A
9E	C6	05	4D	69	B6	83	4F	99	0B	FE	47	67	EB	25	06
B2	BB	B8	D2	03	E1	84	16	BA	A8	ED	9C	20	8E	41	55
05	03	5D	C0	B4	2E	E7	8D	B4	1A	36	23	BC	2F	9F	F3
27	9B	4B	C5	5C	DC	93	60	AC	FC	1A	49	94	E6	B0	0E
5F	24	6E	BB	D8	E3	F3	40	E4	FF	DB	A4	0D	10	CF	57
43	5A	85	51	E4	7D	A5	C9	6D	D5	E9	6A	0E	C6	C5	1F
CF	C6	39	E0	66	1E	17	C7	3D	E0	F3	3D	ED	02	FB	48
D5	43	16	10	E9	CD	1B	9F	1D	07	AC	40	0E	E1	70	8E
4D	E2	56	65	B5	2B	C6	22	D2	12	07	B0	BE	9D	F3	8A
EA	12	26	66	9F	E4	6A	7D	93	6A	38	FB	DB	00	D3	2D
B7	C2	9F	B8	14	93	ED	B1	17	34	F3	6F	89	32	F2	84
1E	DD	91	54	63	CF	B9	DB	55	3B	E4	25	FD	92	FA	45
64	5D	EA	5C	D2	E9	87	A7	20	2C	B3	C7	C9	91	09	48

Mix Column Matrix M mod m_7(x)

03	02	01	01	01	01	01	01	01	01	01	01	01	01	01	01
02	03	02	01	01	01	01	01	01	01	01	01	01	01	01	01
01	02	03	02	01	01	01	01	01	01	01	01	01	01	01	01
01	01	02	03	02	01	01	01	01	01	01	01	01	01	01	01
01	01	01	02	03	02	01	01	01	01	01	01	01	01	01	01
01	01	01	01	02	03	02	01	01	01	01	01	01	01	01	01
01	01	01	01	01	02	03	02	01	01	01	01	01	01	01	01
01	01	01	01	01	01	02	03	02	01	01	01	01	01	01	01
01	01	01	01	01	01	01	02	03	02	01	01	01	01	01	01
01	01	01	01	01	01	01	01	02	03	02	01	01	01	01	01
01	01	01	01	01	01	01	01	01	02	03	02	01	01	01	01
01	01	01	01	01	01	01	01	01	01	02	03	02	01	01	01
01	01	01	01	01	01	01	01	01	01	01	02	03	02	01	01
01	01	01	01	01	01	01	01	01	01	01	01	02	03	02	01
01	01	01	01	01	01	01	01	01	01	01	01	01	02	03	02
01	01	01	01	01	01	01	01	01	01	01	01	01	01	02	03

Appendix 1: Encryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 2.4

F8	7F	71	1B	F2	73	EA	25	B1	83	F3	FA	4E	2E	3D	23
3F	8E	B9	C5	E0	05	A2	C2	35	64	E6	BA	7D	77	8E	D0
34	F8	C9	81	54	C3	39	B5	CE	27	59	8C	7D	F8	FD	B2
51	CF	FC	D7	B1	9F	59	B8	3B	B9	94	4E	60	E9	0A	94
50	25	7F	BC	59	E6	20	85	D2	5F	BF	F9	1D	59	A2	7C
CB	7A	48	D1	B1	97	73	6F	83	13	AF	54	73	EF	24	C2
39	A4	C3	6F	D3	63	9E	78	79	0C	D9	76	2C	63	7E	39
8B	F0	69	45	A7	A9	38	94	BD	B9	9E	40	84	49	EF	B3
48	69	14	C0	D5	96	05	13	60	B7	70	12	0C	54	38	31
BD	5C	20	6B	04	87	68	9B	DC	6D	F3	8E	EA	64	20	EA
B5	F6	9D	C7	78	AE	C1	19	DA	24	79	7A	3E	6A	F1	74
42	CC	FC	DF	C8	A1	9B	8D	00	48	68	27	D4	10	0A	DF
43	58	F0	C3	02	8C	CA	A0	5F	65	B6	1D	DF	1F	4B	66
EB	C9	E7	4E	14	30	CC	80	B5	40	BF	D1	48	3C	71	79
37	07	5A	EC	B4	7B	48	63	C5	85	D2	DF	0A	C8	F0	17
73	26	67	CB	39	0F	A6	2B	89	CE	77	8C	42	9D	F2	71

Round Key 2

00	00	93	B6	FD	E7	2F	34	91	24	D7	74	58	96	9F	C6
80	B9	89	19	CD	9B	ED	E6	CE	B0	29	66	19	1A	67	A0
82	78	DF	CB	73	AC	AF	36	29	40	22	68	C4	A4	57	ED
05	24	A5	71	34	B9	C0	28	0D	D6	62	2F	19	21	EE	1B
76	14	38	FC	E7	56	32	32	E5	2D	90	3B	3F	E2	D9	93
C0	A8	4B	DE	C1	39	DD	02	FC	3A	C9	93	15	28	B5	DF
42	35	E1	BF	2F	69	1F	9C	59	9A	49	D3	B8	7F	56	DA
CD	73	89	39	65	44	B3	11	5C	AB	D8	5E	DB	7B	A7	74
76	C4	D4	7B	FA	74	9F	AE	7F	67	E4	EF	0D	41	7E	A3
6B	6E	46	99	21	AF	32	6A	84	E7	23	5E	A8	9C	27	B9
07	61	AC	EE	44	5E	BB	D6	0E	AA	D4	6C	34	27	70	8B
C8	2E	D0	47	5C	73	A0	18	5F	01	D1	A2	61	C7	B7	AC
58	D8	04	BD	C2	04	88	83	E2	4C	DB	09	1E	61	CE	1D
3C	FB	73	F8	3D	0E	08	14	BA	87	90	F8	09	63	3C	1E
2B	9E	53	26	1E	AF	10	26	C8	7A	CA	DD	29	E6	01	CE
4C	8E	56	53	BB	F1	1C	1D	2B	4F	35	C4	CB	87	B2	E6

State Array 3: Ciphertext

F8	7F	E2	AD	0F	94	C5	11	20	A7	24	8E	16	B8	A2	E5
BF	37	30	DC	2D	9E	4F	24	FB	D4	CF	DC	64	6D	E9	70
B6	80	16	4A	27	6F	96	83	E7	67	7B	E4	B9	5C	AA	5F
54	EB	59	A6	85	26	99	90	36	6F	F6	61	79	C8	E4	8F
26	31	47	40	BE	B0	12	B7	37	72	2F	C2	22	BB	7B	EF
0B	D2	03	0F	70	AE	AE	6D	7F	29	66	C7	66	C7	91	1D
7B	91	22	D0	FC	0A	81	E4	20	96	90	A5	94	1C	28	E3
46	83	E0	7C	C2	ED	8B	85	E1	12	46	1E	5F	32	48	C7
3E	AD	C0	BB	2F	E2	9A	BD	1F	D0	94	FD	01	15	46	92
D6	32	66	F2	25	28	5A	F1	58	8A	D0	D0	42	F8	07	53
B2	97	31	29	3C	F0	7A	CF	D4	8E	AD	16	0A	4D	81	FF
8A	E2	2C	98	94	D2	3B	95	5F	49	B9	85	B5	D7	BD	73
1B	80	F4	7E	C0	88	42	23	BD	29	6D	14	C1	7E	85	7B
D7	32	94	B6	29	3E	C4	94	0F	C7	2F	29	41	5F	4D	67
1C	99	09	CA	AA	D4	58	45	0D	FF	18	02	23	2E	F1	D9
3F	A8	31	98	82	FE	BA	36	A2	81	42	48	89	1A	40	97

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 3.0: Ciphertext

F8	7F	E2	AD	0F	94	C5	11	20	A7	24	8E	16	B8	A2	E5
BF	37	30	DC	2D	9E	4F	24	FB	D4	CF	DC	64	6D	E9	70
B6	80	16	4A	27	6F	96	83	E7	67	7B	E4	B9	5C	AA	5F
54	EB	59	A6	85	26	99	90	36	6F	F6	61	79	C8	E4	8F
26	31	47	40	BE	B0	12	B7	37	72	2F	C2	22	BB	7B	EF
0B	D2	03	0F	70	AE	AE	6D	7F	29	66	C7	66	C7	91	1D
7B	91	22	D0	FC	0A	81	E4	20	96	90	A5	94	1C	28	E3
46	83	E0	7C	C2	ED	8B	85	E1	12	46	1E	5F	32	48	C7
3E	AD	C0	BB	2F	E2	9A	BD	1F	D0	94	FD	01	15	46	92
D6	32	66	F2	25	28	5A	F1	58	8A	D0	D0	42	F8	07	53
B2	97	31	29	3C	F0	7A	CF	D4	8E	AD	16	0A	4D	81	FF
8A	E2	2C	98	94	D2	3B	95	5F	49	B9	85	B5	D7	BD	73
1B	80	F4	7E	C0	88	42	23	BD	29	6D	14	C1	7E	85	7B
D7	32	94	B6	29	3E	C4	94	0F	C7	2F	29	41	5F	4D	67
1C	99	09	CA	AA	D4	58	45	0D	FF	18	02	23	2E	F1	D9
3F	A8	31	98	82	FE	BA	36	A2	81	42	48	89	1A	40	97

Round Key 2

00	00	93	B6	FD	E7	2F	34	91	24	D7	74	58	96	9F	C6
80	B9	89	19	CD	9B	ED	E6	CE	B0	29	66	19	1A	67	A0
82	78	DF	CB	73	AC	AF	36	29	40	22	68	C4	A4	57	ED
05	24	A5	71	34	B9	C0	28	0D	D6	62	2F	19	21	EE	1B
76	14	38	FC	E7	56	32	32	E5	2D	90	3B	3F	E2	D9	93
C0	A8	4B	DE	C1	39	DD	02	FC	3A	C9	93	15	28	B5	DF
42	35	E1	BF	2F	69	1F	9C	59	9A	49	D3	B8	7F	56	DA
CD	73	89	39	65	44	B3	11	5C	AB	D8	5E	DB	7B	A7	74
76	C4	D4	7B	FA	74	9F	AE	7F	67	E4	EF	0D	41	7E	A3
6B	6E	46	99	21	AF	32	6A	84	E7	23	5E	A8	9C	27	B9
07	61	AC	EE	44	5E	BB	D6	0E	AA	D4	6C	34	27	70	8B
C8	2E	D0	47	5C	73	A0	18	5F	01	D1	A2	61	C7	B7	AC
58	D8	04	BD	C2	04	88	83	E2	4C	DB	09	1E	61	CE	1D
3C	FB	73	F8	3D	0E	08	14	BA	87	90	F8	09	63	3C	1E
2B	9E	53	26	1E	AF	10	26	C8	7A	CA	DD	29	E6	01	CE
4C	8E	56	53	BB	F1	1C	1D	2B	4F	35	C4	CB	87	B2	E6

State Array 2.4

F8	7F	71	1B	F2	73	EA	25	B1	83	F3	FA	4E	2E	3D	23
3F	8E	B9	C5	E0	05	A2	C2	35	64	E6	BA	7D	77	8E	D0
34	F8	C9	81	54	C3	39	B5	CE	27	59	8C	7D	F8	FD	B2
51	CF	FC	D7	B1	9F	59	B8	3B	B9	94	4E	60	E9	0A	94
50	25	7F	BC	59	E6	20	85	D2	5F	BF	F9	1D	59	A2	7C
CB	7A	48	D1	B1	97	73	6F	83	13	AF	54	73	EF	24	C2
39	A4	C3	6F	D3	63	9E	78	79	0C	D9	76	2C	63	7E	39
8B	F0	69	45	A7	A9	38	94	BD	B9	9E	40	84	49	EF	B3
48	69	14	C0	D5	96	05	13	60	B7	70	12	0C	54	38	31
BD	5C	20	6B	04	87	68	9B	DC	6D	F3	8E	EA	64	20	EA
B5	F6	9D	C7	78	AE	C1	19	DA	24	79	7A	3E	6A	F1	74
42	CC	FC	DF	C8	A1	9B	8D	00	48	68	27	D4	10	0A	DF
43	58	F0	C3	02	8C	CA	A0	5F	65	B6	1D	DF	1F	4B	66
EB	C9	E7	4E	14	30	CC	80	B5	40	BF	D1	48	3C	71	79
37	07	5A	EC	B4	7B	48	63	C5	85	D2	DF	0A	C8	F0	17
73	26	67	CB	39	0F	A6	2B	89	CE	77	8C	42	9D	F2	71

Inverse Mix Column Matrix M^{-1} mod $m_7(x)$

3A	35	A7	E9	0F	58	79	10	7F	5C	AC	2F	80	C9	B5	09
35	37	01	85	4B	E6	43	28	62	D8	E3	D6	CB	E8	6A	B5
A7	01	62	24	3C	6A	16	B5	0B	7C	98	57	39	1F	E8	C9
E9	85	24	2B	22	81	4A	C6	58	9D	85	50	73	39	CB	80
0F	4B	3C	22	41	BF	77	A4	53	87	E8	76	50	57	D6	2F
58	E6	6A	81	BF	DD	A0	36	AF	D7	1E	E8	85	98	E3	AC
79	43	16	4A	77	A0	07	15	0C	AD	D7	87	9D	7C	D8	5C
10	28	B5	C6	A4	36	15	18	32	0C	AF	53	58	0B	62	7F
7F	62	0B	58	53	AF	0C	32	18	15	36	A4	C6	B5	28	10
5C	D8	7C	9D	87	D7	AD	0C	15	07	A0	77	4A	16	43	79
AC	E3	98	85	E8	1E	D7	AF	36	A0	DD	BF	81	6A	E6	58
2F	D6	57	50	76	E8	87	53	A4	77	BF	41	22	3C	4B	0F
80	CB	39	73	50	85	9D	58	C6	4A	81	22	2B	24	85	E9
C9	E8	1F	39	57	98	7C	0B	B5	16	6A	3C	24	62	01	A7
B5	6A	E8	CB	D6	E3	D8	62	28	43	E6	4B	85	01	37	35
09	B5	C9	80	2F	AC	5C	7F	10	79	58	0F	E9	A7	35	3A

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 2.3

A9	C6	42	E4	37	81	66	DF	18	2B	57	AB	E7	C7	7F	24
4C	A5	53	69	9C	AB	9E	2C	6D	A7	20	E2	CD	3A	46	9E
1E	D3	EA	31	CA	34	CD	0C	F2	84	83	07	AA	C1	DB	9A
9E	C6	05	4D	69	B6	83	4F	99	0B	FE	47	67	EB	25	06
B2	BB	B8	D2	03	E1	84	16	BA	A8	ED	9C	20	8E	41	55
05	03	5D	C0	B4	2E	E7	8D	B4	1A	36	23	BC	2F	9F	F3
27	9B	4B	C5	5C	DC	93	60	AC	FC	1A	49	94	E6	B0	0E
5F	24	6E	BB	D8	E3	F3	40	E4	FF	DB	A4	0D	10	CF	57
43	5A	85	51	E4	7D	A5	C9	6D	D5	E9	6A	0E	C6	C5	1F
CF	C6	39	E0	66	1E	17	C7	3D	E0	F3	3D	ED	02	FB	48
D5	43	16	10	E9	CD	1B	9F	1D	07	AC	40	0E	E1	70	8E
4D	E2	56	65	B5	2B	C6	22	D2	12	07	B0	BE	9D	F3	8A
EA	12	26	66	9F	E4	6A	7D	93	6A	38	FB	DB	00	D3	2D
B7	C2	9F	B8	14	93	ED	B1	17	34	F3	6F	89	32	F2	84
1E	DD	91	54	63	CF	B9	DB	55	3B	E4	25	FD	92	FA	45
64	5D	EA	5C	D2	E9	87	A7	20	2C	B3	C7	C9	91	09	48

Inverse P-box Matrix P_2^{-1} mod $m_7(x)$

C3	30	9F	0C	16	F9	89	A3	AA	95	05	B8	FB	F8	2A	4C
79	71	67	B4	3B	8B	AC	38	02	3A	35	1B	2F	19	CA	8A
A0	51	B3	0A	9C	DA	61	D9	D3	75	04	6A	52	8D	F2	CE
A4	03	68	E5	DF	C2	AE	47	B2	D5	0D	24	59	BB	99	78
B7	E1	A2	14	40	82	EA	D4	BC	CC	06	2D	22	72	4D	48
15	A1	26	42	17	98	FC	4B	F1	70	5C	0E	32	F3	E7	62
6B	AD	BD	00	EE	4A	B6	6E	69	5B	9E	37	CF	B1	CB	87
E3	4F	F7	E8	E4	90	FF	7D	D7	96	C5	B0	01	C4	1D	C1
9B	C0	7F	5A	77	23	25	28	A6	56	E0	1C	C6	3F	A5	84
85	E9	D1	FE	94	DC	AB	27	6F	65	ED	45	C7	FA	DE	3D
8C	43	CD	97	BE	92	60	18	88	EB	74	36	C8	0B	E2	8F
FD	7B	93	A9	53	55	2B	B9	91	F0	58	0F	63	8E	EC	7C
BA	34	A7	81	20	10	41	08	E6	57	AF	D0	A8	5D	83	1A
2C	F5	F4	12	11	54	39	DD	76	BF	3E	3C	5E	73	66	D8
7E	EF	D6	9D	46	50	D2	DB	13	7A	21	09	5F	4E	49	6D
80	33	31	6C	44	9A	86	C9	2E	1F	07	1E	64	29	F6	B5

State Array 2.2

C5	0D	6D	C6	83	57	ED	B3	7D	25	31	E1	E4	FE	23	B0
E4	14	B8	55	D2	05	37	B4	9F	3A	2D	E2	6A	CF	C7	2C
9F	E4	20	7D	47	A5	5D	C7	C9	91	7F	C6	B7	9C	20	CD
C6	EA	BC	5D	12	20	40	49	2C	ED	A7	9C	6F	48	F3	C6
03	6A	C0	43	D2	3D	63	4F	55	FA	DC	8D	24	41	92	24
CF	D3	AA	B5	93	2B	D5	6A	07	67	51	FC	36	00	89	FD
1B	CD	F3	BE	C9	E0	F2	53	05	AC	07	27	5C	45	60	3D
1A	A5	8E	32	AC	84	17	E4	06	4C	3B	E2	8A	40	1E	85
64	66	E1	D3	1F	CF	87	0E	1D	66	9E	AB	D5	C1	9D	8E
E3	D2	CD	56	66	2B	FF	10	2E	25	E9	43	CA	54	1A	42
1E	03	B8	DF	9E	C5	6D	26	DB	65	18	17	9E	9B	83	38
A4	E6	99	EA	69	48	93	B2	AB	22	EA	EB	BA	4B	E9	34
5A	57	B6	A9	10	DB	0E	ED	0E	A7	46	B0	A8	16	9A	94
FB	39	B9	F2	16	0B	91	E4	84	0C	34	DB	1E	B1	FB	69
E9	BB	70	5F	D8	4D	93	9F	BB	C6	84	07	F3	F3	5C	DD
12	B4	DB	2F	9F	C2	09	6E	C7	81	02	E7	E7	4D	E0	F3

Mix Column Matrix M mod $m_6(x)$

03	02	01	01	01	01	01	01	01	01	01	01	01	01	01	01
02	03	02	01	01	01	01	01	01	01	01	01	01	01	01	01
01	02	03	02	01	01	01	01	01	01	01	01	01	01	01	01
01	01	02	03	02	01	01	01	01	01	01	01	01	01	01	01
01	01	01	02	03	02	01	01	01	01	01	01	01	01	01	01
01	01	01	01	02	03	02	01	01	01	01	01	01	01	01	01
01	01	01	01	01	02	03	02	01	01	01	01	01	01	01	01
01	01	01	01	01	01	02	03	02	01	01	01	01	01	01	01
01	01	01	01	01	01	01	02	03	02	01	01	01	01	01	01
01	01	01	01	01	01	01	01	02	03	02	01	01	01	01	01
01	01	01	01	01	01	01	01	01	02	03	02	01	01	01	01
01	01	01	01	01	01	01	01	01	01	02	03	02	01	01	01
01	01	01	01	01	01	01	01	01	01	01	02	03	02	01	01
01	01	01	01	01	01	01	01	01	01	01	01	02	03	02	01
01	01	01	01	01	01	01	01	01	01	01	01	01	02	03	02
01	01	01	01	01	01	01	01	01	01	01	01	01	01	02	03

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 2.1

74	D0	26	E2	9F	32	57	D3	7B	F2	11	96	BE	50	AE	32
37	26	93	F6	07	4C	6A	5E	01	9D	8C	1C	2A	06	2A	3A
49	3C	BB	B0	E2	75	34	E0	A4	AD	A7	BF	A9	4C	7E	41
FA	B0	AF	AA	B7	B8	D5	A0	DC	52	AF	54	A3	C9	52	52
CF	69	8E	2F	C8	F4	AD	92	34	9C	A6	0B	A8	A7	0B	E0
A7	B9	35	3F	27	61	0E	99	1D	C2	08	74	EF	F1	66	F7
58	35	BE	A7	FF	A2	B5	9E	EA	10	B4	1E	B4	B7	D8	17
67	41	1E	81	CD	29	A6	85	C7	F3	1A	22	84	32	9A	3A
D2	A3	33	9B	4C	FC	21	AA	A1	81	4E	C7	A2	80	97	B0
59	3E	45	EA	63	DB	D7	31	36	B9	5C	F3	81	BB	74	78
27	AC	B8	28	01	7F	79	00	DA	35	31	67	64	29	71	0F
95	05	14	DF	6C	21	57	50	CA	37	F1	4D	E6	3F	58	BA
87	FA	38	EB	4E	23	0D	8A	8E	09	29	81	B7	1C	A3	E9
F1	73	CF	AC	58	FC	92	F4	34	4D	E9	1D	DA	B0	DE	EC
3B	17	B1	48	C4	91	99	C9	93	ED	95	38	E6	1F	F7	F1
0E	53	12	ED	B9	03	73	D7	A0	BE	4F	76	D0	B1	85	64

Inverse S-box Matrix S_2^{-1} mod $m_6(x)$

6A	AA	04	4B	88	D1	AF	83	72	23	2F	CC	A8	64	62	27
0E	EC	0A	09	2C	F8	B1	99	D8	38	18	CD	37	6B	50	91
E6	5F	F3	8F	0B	C3	2B	20	13	FC	7B	D5	DA	EE	11	85
3C	B4	47	4A	E2	43	9E	DF	F5	3A	F9	34	3F	82	81	A7
C6	49	BE	DD	03	21	3B	69	79	92	AB	94	EB	E7	74	68
61	C5	4D	63	C8	4E	3E	75	33	F1	BA	E0	D6	E1	D7	8C
12	1B	36	00	59	70	ED	5D	96	97	67	CF	5B	46	17	8A
42	D2	7D	86	0D	D0	8D	C7	9C	EA	40	45	01	DB	C9	1F
BB	05	A2	E5	7F	8B	5C	DC	57	B9	4C	AD	A5	A3	52	E9
3D	0C	B5	87	AC	A6	AE	98	48	28	7A	A9	CB	10	90	F4
60	1D	9F	9B	BC	35	E3	55	2A	41	B8	FE	DE	44	A4	89
2D	5A	F7	1E	CA	39	BD	F0	FD	C0	65	71	B3	54	B0	32
9A	C2	0F	25	B7	66	26	F6	BF	4F	78	8E	95	F2	29	31
D3	9D	D9	1A	93	C4	51	02	58	A1	19	B2	6E	A0	2E	B6
14	1C	24	6C	30	73	EF	CE	08	D4	84	FA	5E	E4	FB	80
06	76	FF	07	15	E8	C1	7E	53	7C	56	16	6F	6D	22	77

State Array 2.0

0D	D3	2B	24	F4	47	75	1A	45	FF	EC	AE	B0	61	A4	47
DF	2B	87	C1	83	EB	67	D7	AA	10	A5	37	7B	AF	7B	F9
92	3F	71	2D	24	D0	E2	14	BC	44	55	32	41	EB	C9	49
56	2D	89	B8	F0	FD	C4	60	6E	4D	89	C8	9B	4F	4D	4D
31	97	52	85	BF	15	44	B5	E2	CB	E3	CC	2A	55	CC	14
55	C0	43	A7	20	1B	62	28	6B	0F	72	0D	80	76	ED	7E
33	43	B0	55	77	9F	39	90	84	0E	CA	50	CA	F0	58	99
5D	49	50	05	F2	FC	E3	8B	F6	07	18	F3	7F	47	7A	F9
D9	9B	4A	A9	EB	6F	5F	B8	1D	05	74	F6	9F	BB	98	2D
F1	81	21	84	00	B2	02	B4	9E	C0	D6	07	05	71	0D	9C
20	DE	FD	13	AA	1F	EA	6A	19	43	B4	5D	59	FC	D2	27
A6	D1	2C	B6	5B	5F	75	61	78	DF	76	E7	EF	A7	33	65
DC	56	F5	FA	74	8F	64	4C	52	23	FC	05	F0	37	9B	D4
76	86	31	DE	33	6F	B5	15	E2	E7	D4	6B	19	2D	2E	5E
34	99	5A	79	B7	0C	28	4F	87	E4	A6	F5	EF	91	7E	76
62	63	0A	E4	C0	4B	86	02	60	B0	68	8D	D3	5A	8B	59

Round Key 1

CE	1D	8E	64	9E	7D	FE	63	69	F7	C0	46	1C	74	7F	97
EC	DE	F0	C0	2F	0F	15	40	69	2F	E7	FE	54	86	7E	C4
4C	DA	B5	85	41	13	7F	9C	BB	FB	29	A3	11	5E	84	88
D0	11	86	AE	7E	AA	04	83	66	76	29	F1	E5	D2	7B	F5
28	75	FC	76	F6	50	F6	04	EF	3E	18	0C	65	2F	11	A4
70	A3	94	10	73	17	28	61	07	5D	F9	E5	10	86	DC	93
39	3D	10	51	E0	CC	4A	2C	88	84	1A	82	16	76	3C	D2
0D	FE	1E	9E	3D	95	0C	32	B4	30	4B	DA	4C	6A	06	72
DA	0D	C3	97	50	41	22	69	D4	A0	C7	58	02	9F	A0	87
4D	03	91	1F	B8	57	B6	1F	A3	45	74	4D	93	A0	02	B2
18	91	AC	5C	BA	98	13	AC	8D	02	5E	4D	13	CC	D2	87
7B	90	F8	E6	7E	F5	DB	72	F4	F0	89	84	72	9C	59	16
C1	E2	8A	08	2E	2F	A8	E5	DB	FC	86	76	BF	4C	24	2F
4A	A9	E1	88	95	E3	DE	27	57	1E	13	D1	89	01	96	74
13	6E	27	2B	8E	86	D7	E3	1E	DF	82	DD	F7	88	E5	18
92	87	56	BA	38	9A	B9	8D	3D	54	EB	07	3D	9B	91	96

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 1.4

C3	CE	A5	40	6A	3A	8B	79	2C	08	2C	E8	AC	15	DB	D0
33	F5	77	01	AC	E4	72	97	C3	3F	42	C9	2F	29	05	3D
DE	E5	C4	A8	65	C3	9D	88	07	BF	7C	91	50	B5	4D	C1
86	3C	0F	16	8E	57	C0	E3	08	3B	A0	39	7E	9D	36	B8
19	E2	AE	F3	49	45	B2	B1	0D	F5	FB	C0	4F	7A	DD	B0
25	63	D7	B7	53	0C	4A	49	6C	52	8B	E8	90	F0	31	ED
0A	7E	A0	04	97	53	73	BC	0C	8A	D0	D2	DC	86	64	4B
50	B7	4E	9B	CF	69	EF	B9	42	37	53	29	33	2D	7C	8B
03	96	89	3E	BB	2E	7D	D1	C9	A5	B3	AE	9D	24	38	AA
BC	82	B0	9B	B8	E5	B4	AB	3D	85	A2	4A	96	D1	0F	2E
38	4F	51	4F	10	87	F9	C6	94	41	EA	10	4A	30	00	A0
DD	41	D4	50	25	AA	AE	13	8C	2F	FF	63	9D	3B	6A	73
1D	B4	7F	F2	5A	A0	CC	A9	89	DF	7A	73	4F	7B	BF	FB
3C	2F	D0	56	A6	8C	6B	32	B5	F9	C7	BA	90	2C	B8	2A
27	F7	7D	52	39	8A	FF	AC	99	3B	24	28	18	19	9B	6E
F0	E4	5C	5E	F8	D1	3F	8F	5D	E4	83	8A	EE	C1	1A	CF

Mix Column Matrix M mod $m_2(x)$

03	02	01	01	01	01	01	01	01	01	01	01	01	01	01	01
02	03	02	01	01	01	01	01	01	01	01	01	01	01	01	01
01	02	03	02	01	01	01	01	01	01	01	01	01	01	01	01
01	01	02	03	02	01	01	01	01	01	01	01	01	01	01	01
01	01	01	02	03	02	01	01	01	01	01	01	01	01	01	01
01	01	01	01	02	03	02	01	01	01	01	01	01	01	01	01
01	01	01	01	01	02	03	02	01	01	01	01	01	01	01	01
01	01	01	01	01	01	02	03	02	01	01	01	01	01	01	01
01	01	01	01	01	01	01	02	03	02	01	01	01	01	01	01
01	01	01	01	01	01	01	01	02	03	02	01	01	01	01	01
01	01	01	01	01	01	01	01	01	02	03	02	01	01	01	01
01	01	01	01	01	01	01	01	01	01	02	03	02	01	01	01
01	01	01	01	01	01	01	01	01	01	01	02	03	02	01	01
01	01	01	01	01	01	01	01	01	01	01	01	02	03	02	01
01	01	01	01	01	01	01	01	01	01	01	01	01	02	03	02
01	01	01	01	01	01	01	01	01	01	01	01	01	01	02	03

State Array 1.3

EE	BB	30	B5	1B	AF	E6	7B	99	4F	C1	2D	1E	15	6C	96
23	F0	F1	53	72	6B	E7	52	7F	E6	2B	E0	6D	A9	65	67
01	A9	E3	00	C8	FB	D5	BC	D7	33	DE	94	3B	F2	45	23
27	49	1C	F7	53	D3	97	BD	D5	B6	96	27	B5	2F	09	92
EA	42	88	33	CE	CF	C3	F3	6D	10	8B	55	9B	07	66	EE
1D	05	5F	41	FF	8A	B1	EA	00	3B	6B	1F	FD	A0	7C	F9
E9	E7	57	4A	EE	A1	6F	07	B1	FA	97	41	35	2F	95	12
D9	6E	5B	11	D1	FD	AA	F5	4F	01	5C	70	CD	A7	96	1B
0F	14	AC	4A	D4	3F	2E	A4	97	82	2A	00	BD	57	6F	95
08	57	B4	CC	E9	99	DD	5F	46	38	AF	AF	3D	85	DC	6C
B1	BC	B2	AA	BC	28	BA	2B	20	3D	33	FD	E1	45	25	D6
E8	E8	39	0E	F0	CA	65	F8	BB	BD	11	C5	02	EF	C0	25
24	BD	4E	87	0A	DD	EE	01	DD	DA	E3	74	E1	8B	82	6C
54	A3	45	25	D2	91	BA	04	DE	C0	6C	60	EE	BA	C9	9A
27	8B	8D	8A	F4	04	64	F0	D1	4F	DB	A6	DA	1E	5A	82
B4	E9	83	7C	80	94	19	85	95	C4	6A	95	C3	B2	4C	5D

Inverse P-box Matrix P_1^{-1} mod $m_2(x)$

46	A6	5E	75	EB	F1	FA	FE	C9	3B	2D	16	E3	A2	C2	40
76	2C	78	BB	13	EA	B3	97	ED	D9	B4	C3	58	29	6E	F8
22	65	FB	E0	06	7D	66	7F	D4	3D	C1	60	9C	2F	79	95
83	34	08	0D	8A	61	6B	20	E2	B5	09	73	19	B9	72	0B
56	45	CA	9F	14	15	7A	C0	42	EE	68	1A	FF	4F	35	E6
BD	A3	6F	F3	1B	80	D3	C7	8B	71	BF	F5	52	AF	27	C8
FC	82	5B	00	0F	33	86	90	AD	43	E1	48	B7	D2	FD	70
2A	DF	8F	24	9E	93	C4	E5	5F	D1	E7	38	01	CF	E9	30
1F	3C	67	F2	2E	C6	0C	CC	07	F6	32	E4	4A	98	4B	D6
CD	5A	3E	CE	8C	BC	0E	64	85	37	C5	7C	EC	F0	55	D5
81	0A	31	A5	A0	23	1E	53	1D	2B	02	8D	17	DA	63	D7
84	D0	74	88	AC	CB	7B	B2	49	21	4D	77	25	DE	4C	51
3A	04	44	A9	5D	62	F7	DD	92	9A	59	10	57	E8	9B	12
89	AB	94	96	99	DB	5C	18	26	39	87	AA	B0	DC	47	1C
D8	A8	A7	B6	EF	11	05	9D	6C	7E	54	41	50	03	BE	91
28	F4	AE	69	A1	6A	BA	B1	36	F9	3F	8E	A4	4E	6D	B8

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 1.2

4A	CD	33	1E	BD	64	C8	97	1C	96	BC	92	2E	F7	DD	EE
74	04	6C	72	CE	CF	2D	E1	04	B5	55	FF	9A	20	BA	0F
BD	BD	01	28	11	02	DE	7C	B4	A9	D9	3D	F0	C1	D4	F2
1B	B2	2A	A1	49	66	95	38	70	C0	24	4F	14	33	B4	6A
96	A6	6D	FA	4E	42	EE	C9	41	BB	BD	6F	C0	11	B2	07
DA	25	FD	2B	DB	DC	EA	E1	6D	E3	57	57	BA	0A	30	4F
94	D3	DD	25	5F	A9	D5	AC	8B	7C	94	97	D1	4C	65	5F
12	3B	09	27	39	B5	23	C5	F1	45	C3	65	AF	FB	4F	BC
8A	B1	E7	27	E8	46	6F	6C	0E	54	53	00	E9	FD	95	5B
07	82	DD	FD	45	23	25	52	57	D2	DA	82	3B	F0	D1	33
BC	80	15	05	C3	AA	BB	8D	8B	87	60	A3	F0	B1	83	A0
EE	85	F8	E7	2B	B6	8A	35	5D	2F	19	53	99	1D	5A	6B
F3	DE	6C	E0	AA	AF	3F	EA	F9	99	88	CA	A4	08	CC	A7
E8	01	2F	B1	D7	6C	95	D6	27	E6	45	91	BA	01	EF	6E
00	97	D5	1E	00	F5	EE	5C	8B	96	6B	1B	3D	7F	10	F4
85	AF	4A	41	E9	1F	82	EE	67	C4	E6	E3	E9	95	7B	9B

State Array 1.1

90	D4	BD	45	3A	3D	BF	C5	BB	C9	78	77	6E	E2	6A	64
76	CB	A7	6E	B0	2D	6C	55	99	8A	D8	06	D9	1E	23	18
48	AB	DA	3D	57	69	12	BB	16	20	DE	ED	01	19	1A	92
24	7A	54	42	CB	A3	C4	0D	AE	42	E7	89	00	59	BE	49
30	81	C4	3C	15	AE	FE	E2	54	6A	7D	ED	63	70	86	C0
3F	17	97	C7	F9	97	2D	B1	B6	11	DF	D1	6C	F9	E9	CE
FC	90	15	BD	05	27	89	24	0D	BA	DF	6C	CA	CE	E5	92
4B	AE	43	10	3D	ED	6E	AA	86	9F	3C	C6	C1	5F	B1	1D
C9	BC	E9	A2	2A	3C	22	EC	00	FC	B3	00	A6	02	C3	EA
13	01	29	D3	47	39	89	98	C3	94	33	0D	D5	1E	64	7B
9F	D4	15	90	7D	CD	31	75	F0	C5	3B	5A	47	50	09	4A
3D	F1	5C	CB	A7	17	D9	1B	BC	19	DD	CB	91	07	D6	6D
43	F8	AC	5D	B0	7B	27	33	B5	B8	46	CB	59	AE	66	C6
BE	03	D0	06	0F	07	08	C1	72	C8	3D	2D	14	DD	F2	8E
86	ED	49	CD	00	9B	43	B8	52	1C	56	A0	A0	C8	BB	12
EA	D0	8D	98	F8	78	37	62	28	3D	9C	2F	48	6E	ED	C4

Inverse Mix Column Matrix M^{-1} mod $m_1(x)$

2B	B0	76	69	5C	65	7E	55	7C	80	01	03	67	C4	62	05
B0	23	97	3A	A1	07	58	94	43	15	A6	CB	D7	4B	BF	62
76	97	5F	31	B2	C7	FB	A6	15	73	84	A1	89	9C	4B	C4
69	3A	31	89	B4	25	1F	A2	4E	A2	1F	25	B4	89	D7	67
5C	A1	B2	B4	A3	59	04	71	93	5A	36	B7	25	A1	CB	03
65	07	C7	25	59	0A	F2	0E	5E	C2	7A	36	1F	84	A6	01
7E	58	FB	1F	04	F2	4D	6E	EC	33	C2	5A	A2	73	15	80
55	94	A6	A2	71	0E	6E	51	FD	EC	5E	93	4E	15	43	7C
7C	43	15	4E	93	5E	EC	FD	51	6E	0E	71	A2	A6	94	55
80	15	73	A2	5A	C2	33	EC	6E	4D	F2	04	1F	FB	58	7E
01	A6	84	1F	36	7A	C2	5E	0E	F2	0A	59	25	C7	07	65
03	CB	A1	25	B7	36	5A	93	71	04	59	A3	B4	B2	A1	5C
67	D7	89	B4	25	1F	A2	4E	A2	1F	25	B4	89	31	3A	69
C4	4B	9C	89	A1	84	73	15	A6	FB	C7	B2	31	5F	97	76
62	BF	4B	D7	CB	A6	15	43	94	58	07	A1	3A	97	23	B0
05	62	C4	67	03	01	80	7C	55	7E	65	5C	69	76	B0	2B

Inverse S-box Matrix S_1^{-1} mod $m_1(x)$

BB	DA	21	9C	CA	9B	31	71	51	33	B5	C2	8E	A6	EB	99
1E	EA	B1	48	B4	89	29	18	D4	2A	1A	76	4F	AA	9D	DF
3E	73	CC	D2	58	3D	A8	05	A1	C9	6E	40	3F	10	17	87
23	BE	26	4D	F5	5E	0B	98	DE	EE	C4	14	B7	6B	A4	36
5D	42	E6	8F	A2	41	B9	80	3C	49	57	9A	83	34	F7	28
1D	E2	B0	7B	67	19	7E	2E	39	30	C1	16	7F	95	45	63
BA	E7	E5	00	78	AE	96	1C	3A	4B	56	0E	72	A9	B8	ED
82	2B	F0	07	68	50	E4	5B	32	5C	D0	E3	01	C6	02	04
AB	CD	88	24	85	44	4E	12	A5	81	03	0C	8D	27	AD	09
FB	0D	7D	3B	CF	94	61	06	F9	DB	DD	FA	E0	C3	D9	91
0F	E8	75	93	C0	C8	F3	6D	5A	F2	D1	4A	D7	65	22	D8
6A	E9	15	F8	C7	BD	D5	8C	B2	70	55	F4	D3	47	86	53
9F	13	7C	B3	66	AF	69	DC	FF	90	46	4C	54	EC	97	A3
BF	62	7A	0A	37	FD	20	25	43	52	BC	1B	9E	11	D6	6C
84	35	5F	E1	92	1F	74	AC	EF	F6	2F	CB	60	2C	64	8B
6F	59	77	38	2D	C5	79	F1	CE	FC	A0	B6	FE	8A	A7	08

Appendix 2: Decryption Step by Step Matrices of Mega Cipher in hexadecimals.

State Array 1.0

FB	37	47	41	C4	6B	53	AF	F4	90	32	5B	B8	5F	56	78
E4	4C	6D	B8	6A	10	72	19	DB	03	43	31	52	9D	D2	D4
3C	4A	BC	6B	2E	4B	B1	F4	29	3E	D6	2C	DA	2A	1A	7D
58	D0	67	E6	4C	93	66	A6	22	E6	AC	81	BB	30	86	49
23	CD	66	B7	89	22	A7	5F	67	56	C6	2C	00	82	4E	9F
36	18	06	DC	FC	06	10	E9	D5	EA	6C	62	72	FC	F6	97
FE	FB	89	47	9B	05	81	58	A6	55	6C	72	46	97	1F	7D
9A	22	8F	1E	6B	2C	B8	D1	4E	91	B7	69	13	63	E9	AA
90	D3	F6	75	6E	B7	CC	60	BB	FE	F8	BB	F3	21	B3	2F
48	DA	C9	0A	80	EE	81	F9	B3	CF	4D	A6	FD	9D	78	E3
91	37	89	FB	C6	EC	BE	50	6F	AF	14	C1	80	1D	33	57
6B	59	7F	4C	6D	18	52	76	D3	2A	11	4C	0D	71	20	A9
8F	CE	D7	95	6A	E3	05	4D	BD	B2	B9	4C	30	22	96	69
86	9C	BF	31	99	71	51	13	F0	FF	6B	10	B4	11	77	AD
4E	2C	49	EC	BB	FA	8F	B2	B0	4F	7E	0F	0F	FF	F4	B1
2F	BF	27	F9	CE	32	98	E5	A1	6B	E0	87	3C	B8	2C	66

Round Key 0

CA	51	29	61	E4	4B	3C	C0	D4	F9	5C	2F	8D	2B	3F	10
CA	6C	19	D9	58	71	1C	7F	EA	6D	63	11	7C	F5	B6	B1
1C	3E	C9	05	00	3F	D0	D4	1D	5F	A8	58	FA	4F	3A	14
1A	B8	15	82	6C	E7	46	D1	0C	92	8C	EE	EF	10	F8	3B
5A	A8	1F	97	DD	43	C6	36	47	33	B2	0C	68	E0	6E	BF
16	38	26	BD	94	65	63	8A	81	CA	04	0C	17	9D	82	F3
8A	82	E8	67	FE	6E	A1	33	CE	3C	09	1D	66	F3	77	18
F2	47	AF	6A	4B	49	D9	B4	2B	FF	CE	49	74	43	8C	CB
F5	B2	82	1D	19	D3	EC	04	C2	98	D8	DA	9C	40	CA	4B
68	A8	A1	65	EF	CE	F3	D9	93	A0	3E	D0	92	EF	58	CF
E5	17	E6	8E	B4	8E	DB	38	0B	DD	60	A0	E4	78	51	77
1E	6B	0A	3F	01	61	21	03	BA	47	64	25	2D	51	55	98
FD	FE	A4	F4	0E	C3	70	20	CE	D3	DD	20	51	43	E4	51
E8	BC	DE	5F	B9	12	3D	72	83	8B	12	3C	DA	77	0E	83
6E	4F	27	88	D2	95	FB	DC	D5	26	5E	2F	6B	8D	D4	91
40	DA	43	D5	BD	40	B8	C9	CC	04	89	B6	1C	D9	58	18

State Array 0: Plaintext

31	66	6E	20	20	20	6F	6F	20	69	6E	74	35	74	69	68
2E	20	74	61	32	61	6E	66	31	6E	20	20	2E	68	64	65
20	74	75	6E	2E	74	61	20	34	61	7E	74	20	65	20	69
42	68	72	64	20	74	20	77	2E	74	20	6F	54	20	7E	72
79	65	79	20	54	61	61	69	20	65	74	20	68	62	20	20
20	20	20	61	68	63	73	63	54	20	68	6E	65	61	74	64
74	79	61	20	65	6B	20	6B	68	69	65	6F	20	64	68	65
68	65	20	74	20	65	61	65	65	6E	79	20	67	20	65	61
65	61	74	68	77	64	20	64	79	66	20	61	6F	61	79	64
20	72	68	6F	6F	20	72	20	20	6F	73	76	6F	72	20	2C
74	20	6F	75	72	62	65	68	64	72	74	61	64	65	62	20
75	32	75	73	6C	79	73	75	69	6D	75	69	20	20	75	31
72	30	73	61	64	20	75	6D	73	61	64	6C	61	61	72	38
6E	20	61	6E	20	63	6C	61	73	74	79	2C	6E	66	79	2E
20	63	6E	64	69	6F	74	6E	65	69	20	20	64	72	20	20
6F	65	64	2C	73	72	20	2C	6D	6F	69	31	20	61	74	7E

A Machine Learning Approach to Predicting Block Cipher Security

Ting Rong Lee[1], **Je Sen Teh**[*1], **Jasy Liew Suet Yan**[1], **Norziana Jamil**[2], and **Wei-Zhu Yeoh**[1]

[1]*School of Computer Sciences, Universiti Sains Malaysia*
[2]*College of Computing and Informatics, Universiti Tenaga Nasional*

E-mail: jesen_teh@usm.my
[*]*Corresponding author*

ABSTRACT

Existing attempts in applying machine learning to cryptanalysis has seen limited success. This paper introduces an alternative approach in applying machine learning to block cipher cryptanalysis. Rather than trying to extract secret keys, machine learning classifiers are trained to predict a cipher's security margin with respect to the number of active s-boxes. Prediction is based on cipher features such as the number of rounds, permutation pattern, and truncated differences. Experiments are performed on a simplified generalised Feistel structure (GFS) block cipher. Prediction accuracy is optimised by refining how cipher features are represented as training data, and tuning hyperparameters. Results show that the machine learning classifiers are able formulate a relationship between the cipher features and security. When used to predict an *unseen* cipher (a cipher whose data was not used for training), an accuracy of up to 62% was obtained, depicting the feasibility of the proposed approach.

Keywords: Active s-box, block cipher, differential cryptanalysis, linear classifier, machine learning, security

1 INTRODUCTION

Encryption algorithms play a vital role in enforcing data confidentiality. These algorithms fall into two categories: asymmetric-key and symmetric-key encryption. Asymmetric-key uses two keys, a private and public key to perform encryption, decryption and authentication tasks whereas symmetric-key algorithms rely on one shared secret key for encryption. As asymmetric-key algorithms are more computationally intensive, their symmetric-key counterparts are preferred to secure communication channels. One widely-used variant of symmetric-key encryption algorithms is the block cipher, which encrypts data in fixed-size blocks. Block ciphers are generally designed based on well-studied structures such as substitution-premutation networks

(SPN), generalised Feistel structures (GFS) or addition-rotation-XOR (ARX). Popular examples of block ciphers include AES, PRESENT (Bogdanov et al., 2007), TWINE (Suzaki et al., 2013) and LBlock (Wu and Zhang, 2011). Block cipher security claims are evaluated in a *trial-by-fire* basis, where cryptanalysts perform attacks on a cipher to identify its weaknesses. Among the various cryptanalytic techniques available, resistance against differential cryptanalysis is one of the de facto security requirements of a block cipher (Biham et al., 2006, Dunkelman and Keller, 2008, Lu et al., 2008a,b, Tsunoo et al., 2008).

Cryptanalysts have explored the use of machine learning as an alternative to conventional cryptanalysis methods. Neural networks have previously been applied to cryptography for various purposes such as strengthening key exchange protocols (Allam et al., 2013), generating s-boxes (Jogdand and Bisalapur, 2011, Kinzel and Kanter, 2002, Klein et al., 2004) or generating encryption keys (Guerreiro and Araujo, 2006, Mandal et al., 2015). Researchers have also looked into the area of adversarial neural cryptography, where the ability of neural networks to communicate securely in the presence of an adversarial network is studied (Coutinho et al., 2018). There were also machine learning-based cryptanalytic research performed on specific ciphers (Alallayah et al., 2012, Alani, 2012, Jain and Mishra, 2018, Mishra et al., 2018). As compared to other cryptanalytic techniques, there is still a lack of research on these machine learning-based approaches.

Our contribution. This work proposes an alternative approach in applying machine learning to cryptanalysis. Rather than trying to predict encryption keys or decrypt ciphertexts, linear classifiers are trained using cipher features to predict the security of a block cipher based on the number of active s-boxes. Given a truncated difference pair along with features such as the number of rounds and permutation pattern, the linear classifiers will determine if a given cipher is secure (the number of active s-boxes is below a certain threshold). Experiments are performed on a simplified GFS cipher as a proof-of-concept, using multiple machine learning classifiers and different representations of the training data to investigate their effect on prediction accuracy. In addition, the ability of the trained classifiers to generalise to unseen cipher variants is examined. Prediction results support the feasibility of the proposed approach, which has the potential to be extended to larger ciphers or various cipher structures.

Outline of the paper. The rest of the paper is structured as follows: Section 2 provides a brief overview of machine learning applications in cryptanalysis. Section 3 details the experimental setup followed by Section 4 which discusses the experimental findings and implications. Section 5 concludes this paper with some final remarks.

2 RELATED WORK

Many researchers have utilised machine learning algorithms, specifically neural networks in the design of encryption algorithms (Allam et al., 2013, Guerreiro and Araujo, 2006, Jogdand and Bisalapur, 2011, Kalaiselvi and Kumar, 2016, Kinzel and Kanter, 2002, Klein et al., 2004, Mandal et al., 2015). In contrast, applications of neural networks specifically to cryptanalyse ciphers are more limited. One of the earliest attempts was the cryptanalysis of simplified DES (SDES), a 12-bit block cipher (Alallayah et al., 2012). Researchers trained a neural network model to emulate the behaviour of SDES. Based on this neuro-model, they were able to successfully determine the secret key corresponding to a plaintext-ciphertext pair. In another study, researchers show that neural networks can map the relationship between plaintext, ciphertext and secret key of SDES (Danziger and Henriques, 2014). However, the performance of the model diminishes in proportion to the strength of the s-box.

A separate attempt using neural networks for cryptanalysis performed known-plaintext attacks on DES and Triple-DES (Alani, 2012). The trained neural network was able to decrypt ciphertexts without knowledge of the secret key. In other words, the neural network was trained to decrypt plaintexts for a specific secret key. Other researchers used sigmoidal neural networks as a cryptanalytic tool for hypothetical Feistel ciphers. However, they noted that their approach is impractical for real world ciphers (Albassal and Wahdan, 2004). More recently, neural networks were used in attempts to cryptanalyse FeW and PRESENT, which are both 64-bit lightweight ciphers (Jain and Mishra, 2018, Mishra et al., 2018). Attacks on both ciphers were based on the same methodology, whereby plaintexts, ciphertexts and intermediate round data corresponding to the same secret key was used for training, validation and testing. The trained neural networks were then used to perform decryption without knowledge of the secret key. However, both attempts were unsuccessful as the trained neural networks have an average accuracy of approximately 50%, which implied that the neural networks were guessing randomly rather than learning the behaviour of the cipher.

Successful attempts to perform concrete attacks (extracting plaintext or keys) have been limited to small scale ciphers such as SDES. As for larger ciphers, the networks have to be trained using datasets generated using the same secret key. The practicality of these approaches is thus limited because multiple secret keys would be used in practice. Rather than trying to perform concrete attacks, this paper investigates the capability of machine learning classifiers in predicting block cipher security in terms of active s-boxes, which can then be extended to resistance against differential attacks. The proposed approach is not only more generalisable to block ciphers based on the same design structure but also will be a useful tool for cryptanalysts or designers in assessing block cipher security.

3 EXPERIMENTAL SETUP

The main goal of the proposed work is to train a machine learning classifier to predict the security level of a block cipher. As security can be defined in a multitude of ways, this work relies on the notions of active s-boxes from the perspective of differential cryptanalysis, which is a de facto standard when it comes to evaluating block cipher security. Differential cryptanalysis is a chosen-plaintext attack in which adversaries are assumed to have gained access to the encryption algorithm and can obtain ciphertexts based on selected plaintexts. Differential cryptanalysis requires the identification of high probability differential trails constructed by the chaining of interconnected differences. A difference between a pair of data blocks, $X' = [X'_0, X'_1, ..., X'_{i-1}]$ and $X" = [X"_0, X"_1, ..., X"_{i-1}]$ is defined as:

$$\Delta X = X' \oplus X" \tag{1}$$

$$\Delta X = [\Delta X_0, \Delta X_1, ..., \Delta X_{i-1}], \tag{2}$$

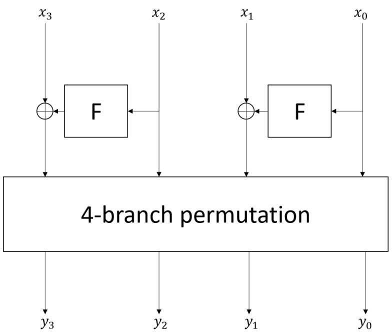


Figure 1: 4-branch GFS (1 round)

where X' and X'' represent the pair of input blocks of the block cipher and Y' and Y'' are their corresponding output blocks. The pair, $\{\Delta X, \Delta Y\}$ is known as a differential pair. For an ideal cipher, given any particular input difference ΔX, the probability of any particular ΔY occurring will be exactly $\frac{1}{2^b}$ where b is the number of bits. Differential cryptanalysis relies on the existence of a differential, $\Delta X \rightarrow \Delta Y$ with a probability far greater than $\frac{1}{2^b}$.

An s-box is defined to be *active* if its input is a non-zero difference. Rather than computing the concrete differential probability for a given differential pair, block ciphers designers provide an estimation of resistance against differential cryptanalysis by calculating the number of active s-boxes. An input difference will be mapped to output differences based on an s-box's differential distribution table. The mapping of differences hold with a certain probability, 2^{-p}. By taking into consideration the best-case (from the attacker's perspective) s-box differential probability, a block cipher is considered to be secure if $2^{AS \times p} \geq 2^b$, where AS denotes the total number of active s-boxes.

As a proof-of-concept the target cipher is a 4-branch GFS cipher. Figure 1 shows one-round of the cipher, which can be repeated multiple times. In the diagram, the notion of truncated differentials is used, whereby x_i denotes a non-zero difference, which can be a 4 or 8 bits depending on the s-box size (Knudsen, 1995). The F function consists of round key addition and the application of one s-box, defined as

$$F(x_i) = s(x_i \oplus rk_i), \tag{3}$$

where s is the s-box function and rk_i is the round key. In essence, the cipher used in the proposed experiments is representative of a 16 or 32-bit block cipher. There are a total of $4! = 24$ possible permutation patterns for a 4-branch permutation.

3.1 Dataset Generation

Data such as the truncated input difference $\hat{X} = \{x_3, x_2, x_1, x_0\}$, truncated output difference $\hat{Y} = \{y_3, y_2, y_1, y_0\}$, number of rounds, r and permutation pattern, P will be used as features. These features were selected because they are generic to any GFS cipher and can be iterated using Matsui's algorithm to generate the dataset. The number of active s-boxes, AS along with a security margin threshold, α will be used to calculate data labels. If $AS > r\alpha$, the input sample is considered to be secure (labelled as 1) whereas if $AS \leq r\alpha$, the input sample is considered to be insecure (labelled as 0). α can be configured based on the desired security margin that the cryptanalyst or designer requires.

In their original specifications, full rounds of block ciphers such as TWINE, PRESENT and LBlock claim to have a security margin of approximately $\alpha = 2$. In the following experiments, we use a lower security bound of $\alpha = 1.5$ because a 4-branch GFS has a maximum of 2 s-boxes per round. Setting $\alpha = 2$ would be a tight bound that very few paths can fulfil (each round would need to have the maximum number of active s-boxes). Thus, $\alpha = 1.5$ would ensure that the security bound is still sufficiently strict, while enabling us to generate enough samples for both secure and insecure labels for training purposes. Sample data for secure and insecure samples are as follows:

- Secure
  - $\hat{X} = \{1, 0, 1, 0\}$
  - $\hat{Y} = \{1, 0, 1, 0\}$
  - $P = \{0, 1, 2, 3\}$
  - $r = 8$
  - $AS = 16$
- Insecure
  - $\hat{X} = \{0, 0, 1, 0\}$
  - $\hat{Y} = \{0, 0, 1, 0\}$
  - $P = \{0, 1, 2, 3\}$
  - $r = 5$
  - $AS = 5$

We exhaustively generate the dataset for all possible input combinations and label them automatically by using an enhanced variant of Matsui's branch-and-bound algorithm (Chen et al., 2017). The algorithm is a variant of the standard branch-and-bound depth-first search algorithm and it can be used to search for differential trails with high probability. The algorithm goes through all possible iterations of differential paths, and then prunes paths with low probabilities that are unlikely to lead to desirable results. However, because no probabilities are involved in the training data, the algorithm was modified to only identify the number of active s-boxes without restriction or a bounding probability.

3.2 Experimental Setup

The supervised learning problem is framed as a binary classification task to determine if a block cipher would be secure or insecure (in terms of the number of active s-boxes) given a specific set of block cipher features. We limit the scope of machine learning algorithms being investigated to linear classifiers, which includes the TensorFlow (TF) linear classifier as well as logistic regression and perceptron from the Python library sklearn. For further optimisation, two representations of the cipher permutation are tested: The first (denoted as rep_1) represents the entire permutation pattern as a single feature whereas the second (denoted as rep_2) represents it as four separate features. For example, the permutation shown in Figure 2 can be represented by rep_1 as $\{0321\}$ or by rep_2 as $\{0, 3, 2, 1\}$, whereby the most significant bit, x_3 is mapped to the least significant bit y_0, the second most significant bit, x_2 is mapped to the most significant bit, y_3 and so on. Experiments are divided into three main phases, for which hyperparameter tuning in

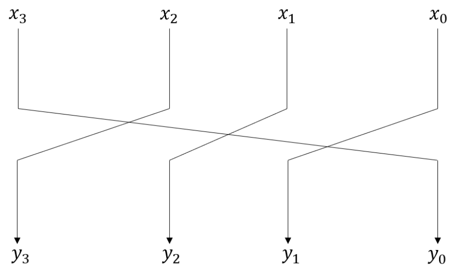


Figure 2: Permutation example

terms of epochs and stopgap iterations were performed to maximise performance:

- **Phase 1 - Baseline Setup** - The preliminary experiments conducted using training and testing data sampled from the entire data pool without any specific conditions. This is to identify which of the machine learning classifiers are most suited for the security margin prediction task as well as the feasibility of the overall approach. A total of 500000 training samples and approximately 130000 testing samples were used, with rep_1 for permutation. The training dataset is balanced, with 50% of the samples labelled as secure/insecure. 5-fold cross validation is used to test the performance of the models, where 5-fold was selected as it leads to test error rates that are neither biased nor have large variance (James et al., 2013, Kuhn and Johnson, 2013). To ensure that the trained linear classifiers are not just randomly guessing, their performance is compared against a random baseline model (sklearn's dummy classifier). The use of the dummy classifier is also to show that a randomly guessing model would not have any advantage in security margin prediction.
- **Phase 2 - Permutation Feature Representation** - Based on the results obtained in Phase 1, the two best linear classifiers are selected to investigate the effect of using rep_1 and rep_2 on prediction accuracy. The same dataset from Phase 1 is used, along with 5-fold cross validation.
- **Phase 3 - Generalisability to Unseen Cipher** - Train-test split is used, where 23 out of the 24 possible permutation patterns are used to generate a training dataset of 250000 samples, whereas the testing data of 138090 samples are generated (exhaustively) from the remaining *unseen* permutation pattern. For this experiment, rep_2 was selected to represent the permutation patterns based on results obtained in Phase 2. This experiment determines if the trained model can generalise to an *unseen* or newly developed cipher.

The performance of the models are evaluated based on the following metrics, in which secure is the positive class and insecure is the negative class:

- **Accuracy**: The total number of true positive and true negative over the total number of examples. In the proposed work's context, this refers to the fraction of predictions that the model has made correctly. From a cryptanalysts' perspective, this reflects how accurately the model can predict whether a given sample is secure or insecure.

Model	Stopgap	Epochs	M-Precision	M-Recall	Accuracy
Dummy	1000	1000	0.50	0.50	0.50
TF Linear Classifier	1000	1000	0.31	0.50	0.62
	500	1000	0.55	0.66	0.49
	750	**750**	**0.70**	**0.51**	**0.63**
	350	750	0.49	0.49	0.62
	500	500	0.48	0.49	0.58
	250	500	0.19	0.50	0.38
Logistic Regression	1000	1000	0.58	0.57	0.52
	500	1000	0.58	0.57	0.52
	750	750	0.58	0.57	0.52
	500	500	0.58	0.57	0.52
	250	500	0.58	0.57	0.52
Perceptron	1000	1000	0.54	0.51	0.58
	500	1000	0.54	0.51	0.58
	300	750	0.54	0.51	0.58
	500	**500**	**0.54**	**0.51**	**0.58**
	250	500	0.54	0.51	0.58

Table 1: Baseline Setup Results

- **Macro-precision**: Precision refers to the number of true positive divided by the total number of true positive and false positive. In the proposed work's context, it refers to the percentage of results that are classified correctly out of the total number of predictions by the model. Macro-precision is the average precision across the two classes.

- **Macro-recall**: Recall refers to the number of true positives divided by the total number of true positives and false negatives. In the proposed work's context, it expresses the percentage of results where the data is classified correctly by the model against the total number of actual examples in a class. of actual data labelled as secure/insecure. Macro-recall is the average recall across the two classes.

4 RESULTS AND DISCUSSION

4.1 Baseline Setup

The baseline setup phase is conducted to depict the trained machine learning classifiers' capability to predict rather than randomly guess the security of a block cipher. All three classifiers outperformed the dummy classifier. However, the best baseline models are the TF linear classifier and perceptron which can both achieve accuracies of 63% and 58% respectively. The hyperparameters that lead to the most accurate prediction results are highlighted in bold in Table 1. This in contrast to the 50% prediction accuracy of dummy classifier which merely guesses the class labels of a balanced dataset. This implies that the linear classifiers are able to formulate a relationship between the cipher features and security.

Model	Permutation	M-Precision	M-Recall	Accuracy
TF Linear Classifier	rep_1	0.70	0.51	0.63
	rep_2	**0.64**	**0.64**	**0.65**
Perceptron	rep_1	0.54	0.51	0.58
	rep_2	**0.64**	**0.52**	**0.63**

Table 2: Comparison results for permutation feature representation

4.2 Permutation Feature Representation

In this section, the two formats for representing permutation patterns are compared. Based on the experimental results in Phase 1, the TF linear classifier and perceptron are selected for testing. The hyperparameter values used in the experiment are the ones highlighted in bold in Table 1. Results for all cases are shown in Table 2. It can be concluded that there is an overall improvement when the permutation pattern is split into four separate features (rep_2) as compared to using a singular feature (rep_1).

4.3 Generalisability to Unseen Cipher

In this phase, the capability of the classifiers in predicting the security of an *unseen* cipher is tested. The classifiers are trained using a dataset generated from 23 out of the 24 possible permutation patterns for a 4-branch GFS, and then tested using a dataset consisting of samples generated from the remaining *unseen* permutation pattern. This would determine if the classifiers can successfully generalize to a block cipher that they have not yet *seen*, or in other words a newly designed cipher. Based on the findings in Phase 2, rep_2 is used to represent the permutation pattern. Results in Table 3 confirm that the TF linear classifier outperforms its peers with a prediction accuracy of 62%. Although perceptron can achieve a slightly higher prediction accuracy of 63%, it performs poorly in terms of macro-recall. The recommended hyperparameter values for the best performing TF linear classifier model are highlighted in bold.

4.4 Discussion

There are many potential practical applications of the proposed work. Firstly, the trained machine learning models can be used alongside existing differential search techniques to narrow down the search space. Even with an accuracy of 62%, such a model would already be useful in this respect. Block cipher designers can also use the trained models to quickly identify if their design could potentially be resistant or weak against differential attacks. As compared to an automated search such as the branch-and-bound algorithm, predictions made by the trained machine learning algorithm for each difference pair is practically instantaneous.

To the best of the authors' knowledge, the proposed methodology of using machine learning to predict block cipher security has yet to be explored. Thus, achieving an accuracy of 62% is a good starting point as it supports the feasibility of the proposed approach. From the machine learning perspective, this implies that the model has been able to learn the relationships between

Model	Stopgap	Epochs	M-Precision	M-Recall	Accuracy
TF Linear Classifier	500	1000	0.30	0.50	0.59
	750	750	0.56	0.67	0.48
	300	750	0.58	0.67	0.50
	500	500	0.53	0.49	0.63
	250	**500**	**0.64**	**0.62**	**0.62**
	100	250	0.21	0.50	0.41
Logistic Regression	500	1000	0.62	0.56	0.54
	750	750	0.62	0.56	0.54
	300	750	0.62	0.56	0.54
	500	500	0.62	0.56	0.54
	250	500	0.62	0.56	0.54
	100	250	0.62	0.56	0.54
Perceptron	500	1000	0.64	0.52	0.63
	750	750	0.64	0.52	0.63
	300	750	0.64	0.52	0.63
	500	500	0.64	0.52	0.63
	250	500	0.64	0.52	0.63
	100	250	0.64	0.52	0.63

Table 3: Unseen Cipher Testing Results

inputs and outputs, leading to more accurate predictions. In contrast, existing work have mostly achieved $\approx 50\%$ accuracy when it comes to predicting secret keys of a cipher directly, which is equivalent to randomly guessing. It sets a precedent for future work in the area, which can involve different block cipher structures, block sizes, feature representations and cryptanalytic attacks. Non-linear machine learning classifiers which could be more suitable for the prediction task can also be investigated since s-boxes have non-linear behaviour. Another possible venue to explore is the potential of machine learning algorithms to be trained using features from smaller scale ciphers to assist in the cryptanalysis of larger, more complex ciphers that share the same structure.

5 CONCLUSION

This paper studied the application of machine learning in cryptanalysis, specifically the use of linear classifiers. Rather than trying to decrypt a ciphertext or extract the secret key, an alternative approach was introduced whereby linear classifiers were trained to predict the security margin of a block cipher. As a proof-of-concept, a 4-branch GFS cipher was used as a target. Three different linear classifiers were trained using cryptanalytic data generated from Matsui's branch-and-bound algorithm, whereby the ratio of the number of active s-boxes to the number of cipher rounds were used as security labels for the training data. By refining how the cipher features are represented, the trained linear classifiers were able to predict the security of the 4-branch GFS cipher with an accuracy of up to 62% on unseen ciphers. This supports the feasibility of the proposed approach, which has a multitude of applications such as filtering differential pairs as starting points for differential cryptanalysis or as a rapid method to determine the security margin of new ciphers.

ACKNOWLEDGMENTS

This work is supported in part by the Ministry of Education Malaysia under the Fundamental Research Grant Scheme (FRGS), project number FRGS/1/2019/ICT05/USM/02/1 and by the Uniten BOLD research grant under Grant No. 10463494/B/2019117.

REFERENCES

Alallayah, K., Amin, M., AbdElwahed, W., and Alhamamii, A. (2012). Applying neural networks for simplified data encryption standard (SDES) cipher system cryptanalysis. In *The International Arab Journal of Information Technology*, pages 163–169. Zarqa University, Jordan.

Alani, M. M. (2012). Neuro-cryptanalysis of DES and triple-DES. In *Neural Information Processing*, pages 637–646. Springer Berlin Heidelberg.

Albassal, A. and Wahdan, A.-M. (2004). Neural network based cryptanalysis of a feistel type block cipher. In *International Conference on Electrical, Electronic and Computer Engineering, 2004. ICEEC '04*. IEEE.

Allam, A. M., Abbas, H. M., and El-Kharashi, M. W. (2013). Authenticated key exchange protocol using neural cryptography with secret boundaries. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Biham, E., Dunkelman, O., and Keller, N. (2006). Related-key impossible differential attacks on 8-round AES-192. In *Topics in Cryptology – CT-RSA 2006*, pages 21–33. Springer Berlin Heidelberg.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. (2007). PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 450–466. Springer Berlin Heidelberg.

Chen, J., Teh, J., Liu, Z., Su, C., Samsudin, A., and Xiang, Y. (2017). Towards accurate statistical analysis of security margins: New searching strategies for differential attacks. *IEEE Transactions on Computers*, 66(10):1763–1777.

Coutinho, M., de Oliveira Albuquerque, R., Borges, F., Villalba, L. G., and Kim, T.-H. (2018). Learning perfectly secure cryptography to protect communications with adversarial neural cryptography. *Sensors*, 18(5):1306.

Danziger, M. and Henriques, M. A. A. (2014). Improved cryptanalysis combining differential and artificial neural network schemes. In *2014 International Telecommunications Symposium (ITS)*. IEEE.

Dunkelman, O. and Keller, N. (2008). An improved impossible differential attack on MISTY1. In *Advances in Cryptology - ASIACRYPT 2008*, pages 441–454. Springer Berlin Heidelberg.

Guerreiro, A. G. and Araujo, C. D. (2006). A neural key generator for a public block cipher. In *2006 Ninth Brazilian Symposium on Neural Networks (SBRN'06)*. IEEE.

Jain, A. and Mishra, G. (2018). Analysis of lightweight block cipher FeW on the basis of neural network. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 1041–1047. Springer Singapore.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer New York.

Jogdand, R. M. and Bisalapur, S. S. (2011). Design of an efficient neural key generation. *International Journal of Artificial Intelligence & Applications*, 2(1):60–69.

Kalaiselvi, K. and Kumar, A. (2016). Enhanced AES cryptosystem by using genetic algorithm and neural network in s-box. In *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*. IEEE.

Kinzel, W. and Kanter, I. (2002). Interacting neural networks and cryptography. In *Advances in Solid State Physics*, pages 383–391. Springer Berlin Heidelberg.

Klein, E., Mislovaty, R., Kanter, I., Ruttor, A., and Kinzel, W. (2004). Synchronization of neural networks by mutual learning and its application to cryptography. In *NIPS*.

Knudsen, L. R. (1995). Truncated and higher order differentials. In *Fast Software Encryption*, pages 196–211. Springer Berlin Heidelberg.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York.

Lu, J., Dunkelman, O., Keller, N., and Kim, J. (2008a). New impossible differential attacks on AES. In *Progress in Cryptology - INDOCRYPT 2008*, pages 279–293. Springer Berlin Heidelberg.

Lu, J., Kim, J., Keller, N., and Dunkelman, O. (2008b). Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1. In *Topics in Cryptology – CT-RSA 2008*, pages 370–386. Springer Berlin Heidelberg.

Mandal, J. K., Datta, D., and Sarkar, A. (2015). Hopfield network based neural key generation for wireless communication (HNBNKG). In *Advances in Intelligent Systems and Computing*, pages 217–224. Springer International Publishing.

Mishra, G., Murthy, S. V. S. S. N. V. G. K., and Pal, S. K. (2018). Neural network based analysis of lightweight block cipher PRESENT. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 969–978. Springer Singapore.

Suzaki, T., Minematsu, K., Morioka, S., and Kobayashi, E. (2013). TWINE: A lightweight block cipher for multiple platforms. In *Selected Areas in Cryptography*, pages 339–354. Springer Berlin Heidelberg.

Tsunoo, Y., Tsujihara, E., Shigeri, M., Saito, T., Suzaki, T., and Kubo, H. (2008). Impossible differential cryptanalysis of CLEFIA. In *Fast Software Encryption*, pages 398–411. Springer Berlin Heidelberg.

Wu, W. and Zhang, L. (2011). LBlock: A lightweight block cipher. In *Applied Cryptography and Network Security*, pages 327–344. Springer Berlin Heidelberg.

Randomness Analysis on RECTANGLE Block Cipher

Abdul Alif Zakaria[*1,3], **A. H. Azni**[*2], **Farida Ridzuan**[2], **Nur Hafiza Zakaria**[1], and **Maslina Daud**[3]

[1]*Faculty of Science and Technology, Universiti Sains Islam Malaysia, Negeri Sembilan 71800, Malaysia*
[2]*CyberSecurity and System Research Unit, Islamic Science Institute (ISI), Universiti Sains Islam Malaysia, Negeri Sembilan 71800, Malaysia*
[3]*CyberSecurity Malaysia, Selangor 63000, Malaysia*

E-mail: alif@cybersecurity.my; ahazni@usim.edu.my;
[*]*Corresponding author*

ABSTRACT

In this paper, we analyze the randomness of the RECTANGLE cipher. RECTANGLE is a lightweight block cipher with 64-bit block size and variants key lengths of 80 and 128 bits. Lightweight block cipher requires less computing power than a block cipher algorithm which makes it more efficient to be implemented in low-resource devices. Randomness is an important property of a cryptography algorithm to make sure the output has no message pattern. The randomness testing was performed using the NIST Statistical Test Suite. A total of nine data categories were applied to generate 1,000 input sequences for each algorithm. RECTANGLE-80 and RECTANGLE-128 passed 98.73% and 98.48% of the randomness tests. Our analysis shows that both RECTANGLE variants seem to be non-random based on the 0.1% significance level. The experimental results from this paper identified some weaknesses that can be addressed in future research.

Keywords: RECTANGLE, block cipher, cryptography, lightweight, statistical test, randomness

1 INTRODUCTION

Low-resource devices like sensor nodes, RFIDs, and smart cards brought notable security issues (Khan and Salah, 2018). Thus, lightweight block cipher gains the attention considering the security offered at a lower cost (Öğünç, 2018). Among the consideration in implementing lightweight algorithms are low energy consumption and high encryption speed (Poschmann, 2009). Since 2011, many algorithms have been developed such as LED (Guo et al., 2011), Piccolo (Shibutani et al., 2011), TWINE (Tomoyasu, 2012), SPARX (Dinu et al., 2016), SIMON and SPECK

(Beaulieu et al., 2015). Security is a huge challenge in low power and lossy networks, so there is a need for further lightweight algorithm development.

RECTANGLE was invented for an embedded system (Zhang et al., 2015). The algorithm acquires a low cost in hardware and efficient in software (Senol, 2017). Though RECTANGLE is highly efficient, its security needs more attention. Improvements on RECTANGLE has been proposed to increase its security (Zhang et al., 2015, Yan et al., 2019). By assessing its security, RECTANGLE can achieve the efficiency and security required for embedded devices.

Randomness test is the techniques which were taken into account during the assessment of the minimum security requirement for a cryptographic algorithm (Ariffin and Yusof, 2017). Statistical analysis of the algorithm may determine if the evaluated cipher meets the security criteria. A non-random block cipher seems to be vulnerable to any type of attack (Isa and Z'aba, 2012). A pseudorandom number generator is critical because the non-authorized user should not be able to guess the cryptographic sequences any easier than a brute force (Chew et al., 2015). Therefore, an algorithm must produce random output. Many algorithms have been analyzed using the NIST Statistical Test Suite such as AES, RC6, Serpent, MARS, and Twofish (Aljohani et al., 2019). Hence, the RECTANGLE must be tested with this method.

The structure of this paper is constructed as follows. A summary of the RECTANGLE is given in Section 2. Next, Section 3 describes the randomness test method. Section 4 addresses the findings and its empirical analysis on RECTANGLE. Finally, Section 5 discusses the conclusion.

2 RECTANGLE BLOCK CIPHER

RECTANGLE contains a block size of 64 bits and accepts 80 or 128 bits key indicated as RECTANGLE-80 and RECTANGLE-128. The encryption runs in 25 rounds using bit-slice methods (Tezcan et al., 2016). RECTANGLE provides excellent performance in software and hardware (Bao et al., 2015, Omrani et al., 2018), which offers flexibility for multiple application platforms.

2.1 Cipher and Subkey States

RECTANGLE presents a cipher state in the form of 4 by 16 array of bits (Feizi et al., 2015). Let $W = w_{63}||\cdots||w_1||w_0$ represent the cipher state. In the first 16 bits, $w_{15}||\cdots||w_1||w_0$ are arranged in $Row(0)$ and the following 16 bits $w_{31}||\cdots||w_{17}||w_{16}$ are located in $Row(1)$ and will continue to do so. Additionally, a 64-bit subkey is employed as 4 by 16 array bits for each round.

2.2 Round Transformation

RECTANGLE operates in substitution-permutation network for a total of 25 rounds. Every round contains three processes including *AddRoundKey*, *SubColumn*, and *ShiftRow*. There is another *AddRoundKey* after the last round. The encryption process for RECTANGLE algorithm is described as follows:

1. *AddRoundKey*: An XOR logical operation of the present state (a) and the round subkey (K).
2. *SubColumn*: Involves column substitution using the RECTANGLE S-box. The input of a S-box is $Col(j) = a_{3,j}||a_{2,j}||a_{1,j}||a_{0,j}$ for $0 \leq j \leq 15$, and the output is $S(Col(j)) = b_{3,j}||b_{2,j}||b_{1,j}||b_{0,j}$. The RECTANGLE S-box operates as a 4-bit to 4-bit S-box, $S : F_2^4 \rightarrow F_2^4$.
3. *ShiftRow*: Every row is left-shifted and rotated on a specific number of positions. $Row(0)$ is remain unchanged. Meanwhile, $Row(1)$, $Row(2)$, and $Row(3)$ are left rotated over 1, 12, and 13 bits respectively.

2.3 Key Expansion

In this section, the RECTANGLE-80 will be used as the illustration. Let $V = v_{79}|| \cdots ||v_1||v_0$ define a key. The 16 rightmost columns of the key are positioned next to each other to establish the 64-bit of the i^{th} subkey K_i at round i. The key register values are updated in every round as follows:

1. Column 0 is rearranged by the S-box, i.e., $k_{3,0}||k_{2,0}||k_{1,0}||k_{0,0} = S(k_{3,0}|| k_{2,0}||k_{1,0}||k_{0,0})$.
2. Applied a 1-round generalized Feistel transformation, i.e., $Row(0) = Row(0) <<< 8 \oplus Row(1)$, $Row(1) = Row(2)$, $Row(2) = Row(3)$, $Row(3) = Row(3) <<< 12 \oplus Row(4)$, and $Row(4) = Row(0)$.
3. $Rc[i]$ is a 5-bits round constant. The 5-bit key state is XOR with $Rc[i]$, i.e.,

$$(k_{4,0}||k_{3,0}||k_{2,0}||k_{1,0}||k_{0,0}) = (k_{4,0}||k_{3,0}||k_{2,0}||k_{1,0}||k_{0,0}) \oplus Rc[i].$$

   Lastly, K_{25} is derived from the revised key state.

3 RANDOMNESS TEST

Analysis on the RECTANGLE was performed on full rounds encryption using the NIST Statistical Suite which comprises of 15 statistical tests with various parameter inputs (Rukhin et al., 2001). The statistical package focuses on various characteristics of non-randomness that may occur in a cipher output.

Eight tests categorize as non-parameterized test selection including Runs (1 p-value), Frequency (1 p-value), Spectral DFT (1 p-value), Binary Matrix Rank (1 p-value), Longest Runs of Ones (1 p-value), Cumulative Sums (2 p-values), Random Excursion (8 p-values), and Random Excursion Variant (18 p-values). The remaining seven tests are categorized as the parameterized test selection which requires the parameter values input. The tests are Block Frequency (1 p-value), Linear Complexity (1 p-value), Maurers Universal (1 p-value), Approximate Entropy (1 p-value), Overlapping Templates (1 p-value), Serial (2 p-values), and Non-Overlapping (148 p-values).

A significance level has to be set to assess the randomness of ciphertext. The significance level, α has to be at least 0.1% (0.001) but not greater than 1%, whereas the minimum sample size is at least the inverse of the significance level ($1 \div 0.001 = 1,000$ samples). If the p-value $\geq \alpha$, the ciphertext is accepted to be random with a 99.9% confidence level (Simion and Burciu, 2019). Conversely, for p-value $< \alpha$, the ciphertext is considered as not random.

In this experiment, the acceptable rejection range of the ciphertext is specified by the confidence interval defined below (Sýs et al., 2015):

$$[p'_a, p'_b] = p' \pm 3\sqrt{\frac{p'(1-p')}{s}} \tag{1}$$

where $p' = 1 - \alpha$, α is the significance level which equals 0.001, and s is the sample size of 1,000 ciphertext. If the number of rejections falls beyond the interval $[p'_a, p'_b]$, then the sample is non-random (Moussaoui et al., 2019).

For a test with one p-value, the acceptable rejection range should be within 0 to 4 samples. Note that Serial and Cumulative Sums produce two p-value each that are tested individually. Same goes to Non-overlapping Template, although the test generates 148 p-values, the p-values are tested individually. Thus, the acceptable rejection range should be within 0 to 4 samples.

Random Excursion (8 p-values) and Random Excursion Variant (18 p-values) may not make use of all 1,000 ciphertext. Some ciphertext might not contain sufficient number of cycles (500 cycles) required for the tests. Thus, the acceptable rejection ranges for both tests differ depending on the samples.

3.1 Data Categories

Nine data categories are used to construct data input in the form of plaintext or key (Abdullah et al., 2011) as shown in Table 1. 1,000 samples are produced using each data category. The blocks number formed in each sample is depending on the block and key sizes (Abdullah et al., 2015). To establish a large bit sequence for the test, the derived blocks are concatenated.

No.	Data Category	RECTANGLE-80				RECTANGLE-128			
		Key	Plaintext	Derived Blocks	Derived Bits	Key	Plaintext	Derived Blocks	Derived Bits
1.	Strict Key Avalanche (SKA) To inspect the sensitiveness of block ciphers to the key bits modifications.	196 random 80-bit keys	All zero	15,680	1,003,520	123 random 128-bit keys	All zero	15,744	1,007,616
2.	Strict Plaintext Avalanche (SPA) To inspect the sensitiveness of block ciphers to the plaintext bit modifications.	All zero	245 random 64-bit plaintext	15,680	1,003,520	All zero	245 random 64-bit plaintext	15,680	1,003,520
3.	Plaintext/Ciphertext Correlation (PCC) To inspect the relation between plaintext and ciphertext pairs using ECB mode of operation.	1 random 80-bit key	15,625 random 64-bit plaintext	15,625	1,000,000	1 random 128-bit key	15,625 random 64-bit plainext	15,625	1,000,000
4.	Ciphertext Block Chaining Mode (CBCM) To inspect the randomness of ciphertext using the CBC mode of operation.	1 random 80-bit key	All zero	15,625	1,000,000	1 random 128-bit key	All zero	15,625	1,000,000
5.	Random Plaintext/Random Key (RPRK) To inspect the randomness of ciphertext using random plaintext and random key.	1 random 80-bit key	15,625 random 64-bit plaintext	15,625	1,000,000	1 random 128-bit key	15,625 random 64-bit plaintext	15,625	1,000,000
6.	Low-Density Key (LDK) To inspect the randomness of ciphertext on the basis of low-density keys.	3,241 specific 80-bit keys	3,241 random 64-bit plaintext	3,241	207,424	3,241 specific 128-bit keys	8,257 random 64-bit plaintext	8,257	528,448
7.	High-Density Key (HDK) To inspect the randomness of ciphertext on the basis of high-density keys.	3,241 specific 80-bit keys	3,241 random 64-bit plaintext	3,241	207,424	3,241 specific 128-bit keys	8,257 random 64-bit plaintext	8,257	528,448
8.	Low-Density Plaintext (LDP) To inspect the randomness of ciphertext on the basis of low-density plaintext.	2,081 random 80-bit keys	2,081 specific 64-bit plaintext	2,081	133,184	2,081 specific 128-bit keys	2,081 random 64-bit plaintext	2,081	133,184
9.	High-Density Plaintext (HDP) To inspect the randomness of ciphertext on the basis of high-density plaintext.	2,081 random 80-bit keys	2,081 specific 64-bit plaintext	2,081	133,184	2,081 specific 128-bit keys	2,081 random 64-bit plaintext	2,081	133,184

Table 1: Data Categories

4 RESULTS AND ANALYSIS

The NIST recommended the input bits for the statistical tests (Rukhin et al., 2001). Runs, Frequency, Block Frequency, and Cumulative Sums require a minimum of 100 bits. Linear Complexity, Random Excursion, Overlapping Templates, and Random Excursion Variant require at least 10^6 bits. Serial, Approximate Entropy, and Non-Overlapping Templates did not specify the bits. Longest Runs of Runs, Spectral DFT, Binary Matrix Rank, and Maurers Universal need at least 128, 1,000, 38,912, and 387,480 bits respectively.

Each data category produced a different length of ciphertext depending on the input as listed in Table 1. SKA, SPA, PCC, CBC, and RPRK can be analyzed through all of the 15 tests (Abdullah et al., 2014). LDP and HDP can only examine ten tests. For LDK and HDK, only ten tests can be examined by RECTANGLE-80, while eleven tests by RECTANGLE-128. This is because LDK, HDK, LDP, and HDP unable to produce sufficient length of data.

Statistical Test	No. of p-value(s)		No. of Samples Evaluated		Range of Acceptable Rejection	
	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128
Runs	1		1,000		[0, 4]	
Frequency						
Spectral DFT						
Block Frequency						
Linear Complexity						
Maurer's Universal						
Binary Matrix Rank						
Approximate Entropy						
Longest Runs of Ones						
Overlapping Templates						
Serial	2					
Cumulative Sums						
Non-Overlapping Templates	148					

Statistical Test			Data Category								
			SKA	SPA	PCC	CBCM	RPRK	LDK	HDK	LDP	HDP
Random Excursion	No. of p-value(s)	REC-80	8								
		REC-128									
	No. of Samples Evaluated	REC-80	635	624	651	625	581	N/A			
		REC-128	645	627	628	622	648				
	Range of Acceptable Rejection	REC-80	[0, 4]	[0, 3]	[0, 4]	[0, 3]	[0, 3]				
		REC-128	[0, 4]	[0, 4]	[0, 4]	[0, 3]	[0, 4]				
Random Excursion Variant	No. of p-value(s)	REC-80	18								
		REC-128									
	No. of Samples Evaluated	REC-80	635	624	651	625	581	N/A			
		REC-128	645	627	628	622	648				
	Range of Acceptable Rejection	REC-80	[0, 4]	[0, 3]	[0, 4]	[0, 3]	[0, 3]				
		REC-128	[0, 4]	[0, 4]	[0, 4]	[0, 3]	[0, 4]				

Table 2: Range of acceptable rejection for RECTANGLE-80 (REC-80) and RECTANGLE-128 (REC-128).

The range of acceptable rejection determines whether a sample pass or fails a test. If the rejected sequences fall within the range, the result is passed. Table 2 shows the evaluated samples for Random Excursion and Random Excursion Variant are less than 1,000 due to insufficient number of cycles. The N/A indicates the test that unable to be performed due to the sample requirement.

	Runs		Frequency		Spectral DFT		Block Frequency		Linear Complexity	
Data Category	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128
SKA	0	0	0	1	0	0	0	0	0	0
SPA	0	0	0	0	0	0	0	0	0	0
PCC	0	0	0	0	0	0	0	0	0	0
CBCM	0	0	0	0	0	0	0	0	0	0
RPRK	0	0	0	0	0	0	0	0	0	0
LDK	0	0	0	0	0	0	0	0	N/A	
HDK	0	0	0	0	0	0	0	0		
LDP	0	0	0	0	0	0	0	0		
HDP	0	0	0	0	0	0	0	0		
	Maurer's Universal		Binary Matrix Rank		Approximate Entropy		Longest Runs of Ones		Overlapping Templates	
Data Category	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128
SKA	0	0	0	0	0	0	0	1	0	0
SPA	0	0	0	0	0	0	0	0	0	0
PCC	0	0	0	0	0	0	0	0	0	0
CBCM	0	0	0	0	0	0	0	0	0	0
RPRK	0	0	0	0	0	0	0	0	0	0
LDK	N/A	0	0	0	0	0	0	0	N/A	
HDK		0	0	0	0	0	0	0		
LDP			0	0	0	0	0	0		
HDP			0	0	0	0	0	0		
	Serial		Cumulative Sums		Non-Overlapping Templates		Random Excursion		Random Excursion Variant	
Data Category	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128	REC-80	REC-128
SKA	0	0	0	2	6	9	0	0	0	0
SPA	0	0	0	0	1	1	0	0	0	0
PCC	0	0	0	0	1	0	0	0	0	0
CBCM	0	0	0	0	1	0	0	0	0	0
RPRK	0	0	0	0	0	1	0	0	1	0
LDK	0	0	0	0	0	3	N/A			
HDK	0	0	0	0	2	1				
LDP	0	0	0	0	3	2				
HDP	0	0	0	0	5	3				

Table 3: Number of rejected p-values for RECTANGLE-80 (REC-80) and RECTANGLE-128 (REC-128).

RECTANGLE-80 passed 13 out of 15 statistical tests. Only two tests failed which are Non-Overlapping Templates and Random Excursion Variant. On the other hand, RECTANGLE-128 passed Runs, Spectral DFT, Block Frequency, Linear Complexity, Maurers Universal, Binary Matrix Rank, Approximate Entropy, Overlapping Templates, Serial, Random Excursion, and Random Excursion Variant tests. RECTANGLE-128 failed Frequency, Longest Runs of Ones, Cumulative Sums, and Non-Overlapping Templates.

Results from Tables 3 indicates that the RECTANGLE did not pass all of the randomness tests. RECTANGLE-80 passed 1,556 out of 1,576 (98.73%) tests and RECTANGLE-128 passed 1,554 out of 1,578 (98.48%) tests. Conclusively, the RECTANGLE is a non-random on the basis of 0.1% significance level.

The overall results show that block cipher algorithm failed most of the tests in the SKA data category with 6 (RECTANGLE-80) and 13 (RECTANGLE-128) fails. SKA is much determined by the sensitiveness of a cipher to changes in the key. The finding shows that a factor that contributed to the results is the weakness of the key schedule method. These results proved that there is a need to improve the RECTANGLE key schedule (Yan et al., 2019).

Another finding points out both RECTANGLE variants mostly failed the Non-Overlapping Templates test with 19 (RECTANGLE-80) and 10 (RECTA-NGLE-128). It shows that the output produced from the algorithm has too many occurrences of a given non-periodic pattern.

Hence, it is important to improve RECTANGLE encryption.

5 CONCLUSION

One of the significant criteria for developing an encryption algorithm is the algorithm's ability to behave as a random number generator. A statistical analysis is capable to determine if the algorithm fulfills this condition. The randomness of the RECTANGLE has been tested and the results show that the algorithm is not random based on the 0.1% significance level using 1,000 samples. An algorithm that passes all of the statistical tests does not guarantee its security (Isa and Z'aba, 2014). However, a secure algorithm should pass all of the tests. For security purposes, enhancement on the RECTANGLE is suggested in the future to improve its security.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) and CyberSecurity Malaysia (CSM) for the support and facilities provided. This research paper is supported by Fundamental Research Grants No: FRGS/1/2019/ICT03/USIM/02/1.

REFERENCES

Khan, M. A., and Salah, K. (2018). IoT security: review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82: 395-411.

Öğünç, M. (2018). Differential cryptanalysis on LBLOCK using differential factors. (Master's thesis).

Poschmann, A. Y. (2009). Lightweight cryptography: cryptographic engineering for a pervasive world. (Ph.D thesis).

Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. (2011). The LED block cipher. *In International Workshop on Cryptographic Hardware and Embedded Systems*, 326-341. Springer, Berlin, Heidelberg.

Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T. (2011). Piccolo: An ultra-lightweight blockcipher. *In International Workshop on Cryptographic Hardware and Embedded Systems*, 342-357. Springer, Berlin, Heidelberg.

Tomoyasu, S. (2012). Twine: A lightweight block cipher for multiple platforms. *In Selected Areas in Cryptography*, 7707. Springer Berlin Heidelberg.

Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., and Verbauwhede, I. (2015). RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12), 1-15.

Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Groschdl, J., and Biryukov, A. (2016). Design strategies for ARX with provable bounds: SPARX and LAX (full version). *IACR Cryptology ePrint Archive*, 2016, 984.

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2015). The SIMON and SPECK lightweight block ciphers. *In Proceedings of the 52nd Annual Design Automation Conference*, 1-6.

Ariffin, S., and Yusof, N. A. M. (2017). Randomness analysis on 3D-AES block cipher. *In 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, 331-335. IEEE.

Isa, H., and Z'aba, M. R. (2012). Randomness analysis on LED block ciphers. *In Proceedings of the Fifth International Conference on Security of Information and Networks*, 60-66.

Senol, A. (2017). Improved differential attacks on RECTANGLE. (Master's thesis).

Bao, Z., Luo, P., and Lin, D. (2015). Bitsliced implementations of the PRINCE, LED and RECTANGLE block ciphers on AVR 8-bit microcontrollers. *In International Conference on Information and Communications Security*, 18-36. Springer, Cham.

Omrani, T., Rhouma, R., and Sliman, L. (2018). Lightweight cryptography for resource-constrained devices: a comparative study and RECTANGLE cryptanalysis. *In International Conference on Digital Economy*, 107-118. Springer, Cham.

Zhang, W., Bao, Z., Rijmen, V., and Liu, M. (2015). A New classification of 4-bit optimal s-boxes and its application to PRESENT, RECTANGLE and SPONGENT. *In International Workshop on Fast Software Encryption*, 494-515. Springer, Berlin, Heidelberg.

Yan, H., Luo, Y., Chen, M., and Lai, X. (2019). New observation on the key schedule of RECTANGLE. *Science China Information Sciences*, 62(3): 32108.

Chew, L. C. N., Shah, I. N. M., Abdullah, N. A. N, Zawawi, N. H. A., Rani, H. A., and Zakaria, A. A. (2015). Randomness analysis on Speck family of lightweight block cipher. *International Journal of Cryptology Research*, 5(1): 44-60.

Aljohani, M., Ahmad, I., Basheri, M., and Alassafi, M. O. (2019). Performance analysis of cryptographic pseudorandom number generators. *IEEE Access*, 7: 39794-39805.

Tezcan, C., Okan, G. O., enol, A., Doan, E., Yceba, F., and Baykal, N. (2016). Differential attacks on lightweight block ciphers PRESENT, PRIDE, and RECTANGLE revisited. *In International Workshop on Lightweight Cryptography for Security and Privacy*, 18-32. Springer, Cham.

Feizi, S., Nemati, A., Ahmadi, A., and Makki, V. A. D. (2015). A high-speed FPGA implementation of a bit-slice ultra-lightweight block cipher, RECTANGLE. *In 2015 5th International Conference on Computer and Knowledge Engineering*, 206-211. IEEE.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., and Barker, E. (2001). A statistical test suite for random and pseudorandom number generators for cryptographic applications. *Booz-allen and hamilton inc mclean va.*

Simion, E., and Burciu, P. (2019). A note on the correlations between NIST cryptographic statistical tests suite. *University Politehnica of Bucharest Scientific Bulletin-Series A-Applied Mathematics and Physics*, 81(1): 209-218.

Sýs, M., Ríha, Z., Matyás, V., Marton, K., and Suciu, A. (2015). On the interpretation of results from the NIST statistical test suite. *Romanian Journal of Information Science and Technology*, 18(1): 18-32.

Moussaoui, S., Zeghdoud, S., and Allailou, B. (2019). Implementation and statistical tests of a block cipher algorithm MISTY1. *Malaysian Journal of Computing and Applied Mathematics*, 2(2): 44-59.

Abdullah, N. A. N., Lot, Zawawi, N. H. A., and Rani, H. A. (2011). Analysis on lightweight block cipher, KTANTAN. *In 2011 7th International Conference on Information Assurance and Security*, 46-51. IEEE.

Abdullah, N. A. N., Chew, L. C. N., Zakaria, A. A., Seman, K., and Norwawi, N. M. (2015). The comparative study of randomness analysis between modified version of LBlock block cipher and its original design. *International Journal of Computer and Information Technology*, 4(6): 867-875.

Abdullah, N. A. N., Seman, K., and Norwawi, N. M. (2014). Statistical analysis on LBlock block cipher. *In International Conference on Mathematical Sciences and Statistics 2013*, 233-245. Springer, Singapore.

Isa, H., and Z'aba, M. R. (2014). Randomness of the PRINCE block cipher. *International Conference on Frontiers of Communications, Networks and Applications*.

Collisions in Block Ciphers: Application to Small-Scale AES Variants

Muhammad Reza Z'aba[*], **Muhammad Akmal Hakim Mohd Zuki**, **Ainuddin Wahid Abdul Wahab**, and **Miss Laiha Mat Kiah**

Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya

E-mail: reza.zaba@um.edu.my, akmalhakimakmalhakim@gmail.com, ainuddin@um.edu.my, misslaiha@um.edu.my
[*]Corresponding author

ABSTRACT

A block cipher should act as a permutation. For a given secret key, two different input plaintext blocks would never result in a collision, i.e. be encrypted to the same ciphertext. This article investigates whether or not two different secret keys would yield a collision (i.e. encrypted to the same ciphertext). Such collisions may have negative effects on hash function constructions based on block ciphers such as Davies-Meyer and Matyas-Meyer-Oseas where the block cipher key is the input message to the hash function. In order to find such collisions, we conducted several experiments on small scale variants of the block cipher AES.

Keywords: block cipher, permutation, collision, encryption

1 INTRODUCTION

A block cipher is a cryptographic primitive that accepts, as inputs, a fixed-length plaintext block and secret key, and outputs a fixed-length ciphertext block. Block ciphers have been used to construct important cryptographic schemes that provide either confidentiality, integrity, data-origin authenticity[1], or all of these security objectives. Rogaway (2011) provided a comprehensive analysis of block cipher modes of operation that provide these objectives via encryption (confidentiality), MAC (authenticity) and authenticated encryption (confidentiality and authenticity) schemes.

To provide integrity, a block cipher can be used to construct hash functions using modes such as Davies-Meyer (DM) and Matyas-Meyer-Oseas (MMO) (Matyas et al. (1985)). In DM,

[1] Authenticity includes integrity.

the block cipher key input is replaced by the user-controlled message. In MMO, on the other hand, the block cipher key input is replaced by the output of the previous iteration of the block cipher. In both cases, the block cipher key is *not* fixed and its value is *known*. Additionally, the key may be under the control of an attacker. In standard confidentiality modes such as cipher block chaining (CBC), for a particular message, the key is *fixed* and its value is *secret*.

For a randomly chosen value of the secret key, a block cipher is a permutation. Two different plaintext blocks would never yield the same ciphertext block for any fixed key. However, for a randomly chosen value of the plaintext block, does the block cipher still act as a permutation if the secret key varies? Stated differently, for a fixed value of the plaintext block, is it possible for two or more distinct secret key values to yield the same ciphertext value?

Knudsen and Rijmen (2007) introduced the notion of known-key security of block ciphers. The assumption is that an attacker has access to, or control over, the secret key. Subsequent works (e.g. Andreeva et al. (2014), Cogliati and Seurin (2015, 2016)) focus on modelling certain components in the block cipher as a pseudorandom function and investigate the minimum number of rounds that the whole block cipher emulates a random permutation. For instance, in the work of Guo and Lin (2015), the SIMON block cipher's round function is assumed to be a public random permutation and that 21 rounds are sufficient for the whole cipher to behave as a random permutation. In the context of hash functions, Mennink and Preneel (2015) presented the weak cipher model, which is a generic model applicable to block ciphers that are amenable to known key attacks.

A related line of work also examines the minimum number of rounds that a particular type of block ciphers emulates a random permutation. However, the investigation is not within the context of known-key distinguishers. Nevertheless, similar assumptions were made to the components of the block ciphers, particularly regarding the round functions and the subkeys. There are works that assume both are independently derived (e.g. Chen and Steinberger (2014)), or only the former is independently derived (e.g. Andreeva et al. (2013), Chen et al. (2014), Dai and Steinberger (2016)).

In this article, we examine the security of block ciphers by using *concrete* instantiation of the components, i.e. by not putting any assumptions on the components as what has been done mostly in the literature, as previously discussed. This is achieved by analysing the permutation property of MINIAES, which is a miniaturised version of the block cipher AES. We will focus on finding collisions in MINIAES. If a collision is found in the cipher, then it is not a permutation.

2 DESCRIPTION OF MINIAES

MINIAES was proposed by Phan (2002) as a small scale variant of the AES (Daemen and Rijmen (1998, 2002)). MINIAES supports 16-bit block and 16-bit key lengths. The cipher originally consists of only two rounds. In this article, we slightly modify the cipher so that it can accept greater number of rounds. The cipher has the same four operations as the AES, but in a miniaturized form.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$s(x)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Table 1: S-box of MINIAES

The state is depicted as a 2×2 matrix where each element is a 4-bit word. The plaintext is initially placed in the matrix column-wise, starting from the left column. The plaintext is then XORed with the first subkey, which is the same as the secret key. Then, the result is subjected to a round function that consists of the following operations. `NibbleSub` substitutes each word based on a 4×4 s-box given in Table 1. `ShiftRow` swaps the 2 words at the second row of the state. `MixColumn` applies the following 2×2 matrix to each of the two columns of the state matrix.

$$\begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}.$$

`KeyAddition` XORs the state with the current round subkey. The last round function omits `MixColumn`.

3 EXPERIMENTAL METHOD

For each cipher, we define four different variants that will be analysed to detect collisions:

1. MINIAES. This is the original variant where the subkeys are generated according to the key schedule algorithm.
2. MINIAES-A. For this variant, each subkey equals the secret key.
3. MINIAES-B. The value of both the first subkey equals the secret key while the value of the last subkey is generated randomly. The remaining subkeys are set to all zeros. This can be seen as an Even-Mansour (EM) construction (Even and Mansour (1993, 1997)).
4. MINIAES-C. The value of both the first and last subkeys equal the secret key. This can be seen as an instance of the single-key Even-Mansour (SEM) construction proposed by Dunkelman et al. (2012).

Let $|x|$ denote the length of x in bits. Each of the above variant is run through the following steps, each time with different number of rounds.

1. Set both the values of the plaintext m, and the counters N_K, N_P, n_i and i to all-zero. The relevance of these counters is described in the next section. Set the Boolean variable n_{col} to false. For our case, let $|m| = |k| = 16$.
2. Encrypt plaintext m with an all-zero secret key, i.e. $k = 0$ and store the corresponding ciphertext, i.e. $c_0 = E(k, m)$.

3. Increment the value of current secret key by 1, i.e. $k = (k + 1)$, and encrypt the plaintext m with the updated value of the key, i.e. $c_1 = E(k, m)$.
4. Compare c_0 with c_1. If both are equal, meaning that a collision occurs, then increment the counter n_i by 1, i.e. $n_i = (n_i + 1)$. Set $n_{\textsf{col}}$ to $\textsf{true}$.
5. If we have reached the end of the key space, i.e. $k = (2^{|k|} - 1)$, then reset the key to zero, i.e. $k = 0$ and proceed to the next step, else repeat Step 3.
6. Add the current value of the counter N_K with n_i, i.e. $N_K = (n + n_i)$ and increment the counter i by 1, i.e. $i = (i + 1)$.
7. Increment the plaintext m by 1, i.e. $m = (m + 1)$. If $n_{\textsf{col}}$ equals $\textsf{true}$, then increment the counter N_P by 1, i.e. $N_P = N_P + 1$. Reset the value $n_{\textsf{col}}$ to $\textsf{false}$. Repeat Step 2 until we have reached the end of the plaintext space, i.e. $m = (2^{|m|} - 1)$.

In the experiment above, for each unique plaintext, we only compare the encryption with an all-zero key with the encryption with other secret keys. A comprehensive analysis should compare not only with an all-zero key, but with all possible keys. Since such task requires an enormous computation (despite using a small scale AES), we did not pursue this path.

4 RESULTS AND DISCUSSION

The results of the experiment stated in Section 3 are provided in Table 2 for MINIAES. For each variant and round number, the table provides the number of colliding pairs, the total number of secret keys that produce collisions, the total number of different plaintexts that cause collisions and the ratio of keys to plaintexts.

For each plaintext, the number of colliding pairs is counted as

$$\binom{n_i}{2} = \frac{n_i!}{2(n_i - 2)!} = \frac{n_i \cdot (n_i - 1)}{2}$$

where n_i is the number of distinct keys that yield a collision as in Step 4 of our experiment. Assume that when $i = 0$, i.e. when $m = \texttt{0000}$, we obtained the following collisions:

$$E(k_a, m) = E(k_b, m) = E(k_c, m) = E(k_d, m) = E(k_e, m).$$

For the above keys, there are $\binom{5}{2} = 10$ colliding pairs, i.e. $(E(k_a, m), E(k_b, m))$, $(E(k_a, m), E(k_c, m))$, ..., $(E(k_a, m), E(k_e, m))$, $(E(k_b, m), E(k_c, m))$, ..., and $(E(k_d, m), E(k_e, m))$.

The number of colliding pairs N_C given in the table is the sum of colliding pairs for all plaintexts, i.e.

$$N_C = \sum_{i=0}^{2^{|m|}-1} \left(\frac{n_i \cdot (n_i - 1)}{2} \right).$$

Variant	Rounds r	Colliding pairs (N_C)	Secret keys (N_K)	Plain- texts (N_P)	Keys per plaintext
MiniAes	2	97,900	106,599	41,290	2.58
	4	99,383	107,502	41,528	2.59
	6	97,444	106,123	41,185	2.58
	8	98,473	107,081	41,441	2.58
MiniAes-a	2	78,965	96,930	39,670	2.44
	4	98,750	107,197	41,440	2.59
	6	98,376	106,989	41,423	2.58
	8	98,492	107,205	41,547	2.58
MiniAes-b	2	98,750	107,004	41,327	2.59
	4	98,799	197,225	41,507	2.58
	6	97,854	106,643	41,298	2.58
	8	98,330	107,191	41,557	2.58
MiniAes-c	2	575,168	231,863	55,335	4.19
	4	100,542	107,111	41,495	2.58
	6	109,801	109,971	41,639	2.64
	8	109,782	110,690	41,744	2.65

Table 2: Results for MiniAes

The number of secret keys N_K in our experiment (see Step 6) does not represent *unique* keys. There is a possibility of counting the same key multiple times. For instance, there may exists the following collisions for two different plaintexts m_0 and m_1:

$$\begin{aligned} E(k_a, m_0) = E(k_b, m_0) = E(k_c, m_0) = E(k_d, m_0) \\ E(k_a, m_1) = E(k_e, m_1) = E(k_d, m_1) = E(k_f, m_1) \end{aligned}$$

In the above example, the keys k_a and k_d are counted twice and based only on the above example, $N_K = 8$. Therefore, it is possible for the value of N_K to exceed $2^{|k|} - 1$.

The number of plaintexts N_P refer to the total number of unique plaintexts that yield the same ciphertexts if encrypted using certain secret keys. For instance, assume that we have the following set of collisions:

$$\begin{gathered} E(k_a, m_0) = E(k_b, m_0) = E(k_c, m_0) = E(k_d, m_0) \\ E(k_a, m_1) = E(k_e, m_1) = E(k_d, m_1) = E(k_f, m_1) \\ E(k_g, m_2) = E(k_h, m_2) \\ E(k_d, m_3) = E(k_i, m_3) \end{gathered}$$

In the above example, there are four unique plaintexts and hence, $N_P = 4$.

As depicted in Table 2, collisions occur for all MiniAes variants and as early as 2 rounds. If we run the experiments for more than the rounds stated in the table, collisions would still occur. The results for these additional rounds are similar to the ones in Table 2 with the exception of MiniAes-c where collisions for 2 rounds are significantly higher than 4 or more rounds. The results for MiniAES variants greater than 8 rounds are omitted for brevity. Therefore, adding more rounds do not have a significant effect in reducing the number of collisions.

With the exception of MINIAES-C, the number of keys per plaintext as the number of rounds increases stood at 2.58. On average, for each plaintext, there are two distinct keys that yield a collision. In contrast, if the experiments were done by fixing the key and varying the plaintexts, there would be no collisions, even for 1 round. This is because the round function is a permutation if the key is fixed. Our experiments show that the round function is not a permutation if we fix the plaintext and vary the keys.

As stated in Section 3, MINIAES-C is an instance of the SEM construction proposed by Dunkelman et al. (2012). The construction aims to further simplify the original EM construction which requires the key length to be twice the block length (Even and Mansour (1993)). The SEM construction allows the key length to be equal to the block length, while providing similar security level to that of the original EM construction. Interestingly, as shown in Table 2, 2-round MINIAES-C generates 5 times more colliding pairs than the other MINIAES variants, including MINIAES-B, which is an instance of the EM construction. In fact, the number of keys per plaintext is also slightly higher than the other variants at 6 rounds and greater.

In the Davies-Meyer scheme, the input to be hashed is subjected to the key schedule of the underlying block cipher. The message input of the block cipher is the previous hash value, or a fixed initial value. The key input of the block cipher is therefore under the control of an adversary. If a collision can be obtained for the block cipher, then it may be translated into a collision for the hash function.

Note that our analysis is limited to performing encryption of one block of plaintext (i.e. ECB mode). In real-world applications, the length of a plaintext typically exceeds the block length. Furthermore, the encryption is supposed to be performed using other modes than ECB (e.g. CBC, CTR).

5 CONCLUSION

We have empirically shown that all four variants of MINIAES considered in this article are not a permutation if the plaintext is fixed and the keys are varied. The four variants analyzed are MINIAES where: (1) the key schedule is present, (2) the subkeys equal the secret key, (3) the EM construction is employed, (4) the SEM construction is used. The results may carry over to the full version of the AES. However, finding different keys that yield a collision for AES is a challenge due to the block length and key length of the cipher.

A block cipher is a permutation if the key is fixed and the plaintext is varied. It is an open problem whether a block cipher can be constructed where it is a permutation, i.e. collision-free, even though the plaintext is fixed and the keys are varied.

ACKNOWLEDGMENTS

This work was supported by University of Malaya's Research University Grant – Faculty Programme, Grant Number GPF004D-2018.

REFERENCES

Andreeva, E., Bogdanov, A., Dodis, Y., Mennink, B., and Steinberger, J. P. (2013). On the Indifferentiability of Key-Alternating Ciphers. In Canetti, R. and Garay, J. A., editors, *Advances in Cryptology — CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 531–550. Springer-Verlag.

Andreeva, E., Bogdanov, A., and Mennink, B. (2014). Towards Understanding the Known-Key Security of Block Ciphers. In Moriai, S., editor, *Fast Software Encryption, FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 348–366. Springer-Verlag.

Chen, S., Lampe, R., Lee, J., Seurin, Y., and Steinberger, J. P. (2014). Minimizing the Two-Round Even-Mansour Cipher. In Garay, J. A. and Gennaro, R., editors, *Advances in Cryptology – CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 39–56. Springer-Verlag.

Chen, S. and Steinberger, J. P. (2014). Tight Security Bounds for Key-Alternating Ciphers. In Nguyen, P. Q. and Oswald, E., editors, *Advances in Cryptology – Eurocrypt 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer-Verlag.

Cogliati, B. and Seurin, Y. (2015). On the Provable Security of the Iterated Even-Mansour Cipher Against Related-Key and Chosen-Key Attacks. In Oswald, E. and Fischlin, M., editors, *Advances in Cryptology — EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 584–613. Springer-Verlag.

Cogliati, B. and Seurin, Y. (2016). Strengthening the Known-Key Security Notion for Block Ciphers. In Peyrin, T., editor, *Fast Software Encryption, FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 494–513. Springer-Verlag.

Daemen, J. and Rijmen, V. (1998). AES proposal: Rijndael. NIST AES Proposal.

Daemen, J. and Rijmen, V. (2002). *The Design of Rijndael, AES – The Advanced Encryption Standard*. Springer-Verlag.

Dai, Y. and Steinberger, J. P. (2016). Indifferentiability of 8-Round Feistel Networks. In Robshaw, M. and Katz, J., editors, *Advances in Cryptology — CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 95–120. Springer-Verlag.

Dunkelman, O., Keller, N., and Shamir, A. (2012). Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In Pointcheval, D. and Johansson, T., editors, *Advances in Cryptology – Eurocrypt 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 336–354. Springer-Verlag.

Even, S. and Mansour, Y. (1993). A Construction of a Cipher From a Single Pseudorandom Permutation. In Imai, H., Rivest, R. L., and Matsumoto, T., editors, *Advances in Cryptology – ASIACRYPT '91*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer-Verlag.

Even, S. and Mansour, Y. (1997). A Construction of a Cipher From a Single Pseudorandom Permutation. *Journal of Cryptology*, 10(3):151–161.

Guo, C. and Lin, D. (2015). On the Indifferentiability of Key-Alternating Feistel Ciphers with No Key Derivation. In Dodis, Y. and Nielsen, J. B., editors, *Theory of Cryptography Conference – TCC 2015, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 110–133. Springer-Verlag.

Knudsen, L. R. and Rijmen, V. (2007). Known-key distinguishers for some block ciphers. In Kurosawa, K., editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer-Verlag.

Matyas, S. M., Meyer, C. H., and Oseas, J. (1985). Generating Strong One-Way Functions with Cryptographic Algorithm. *IBM Technical Disclosure Bulletin*, 27(10A):5658–5659.

Mennink, B. and Preneel, B. (2015). On the Impact of Known-Key Attacks on Hash Functions. In Iwata, T. and Cheon, J. H., editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 59–84. Springer-Verlag.

Phan, R. C.-W. (2002). Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students. *Cryptologia*, 26(4):283–306.

Rogaway, P. (2011). Evaluation of Some Blockcipher Modes of Operation. CRYPTREC Report.

Comparison Analysis of Cryptographic Electronic Systems

Yun-Xing Kho[1] and **Swee-Huay Heng**[*1]

[1]*Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia*

E-mail: yunxing_535357@hotmail.com
[*]*shheng@mmu.edu.my*

ABSTRACT

In this digital age, there have been a lot of cryptographic electronic systems proposed in the literature to ease human life. In this paper, we compare the cryptographic techniques used in electronic voting systems, electronic auction systems, electronic cash systems, and electronic cheque systems. We analyse the structure, security requirements, and the underlying tools or schemes used in the respective electronic systems. We also take a closer look to analyse all the existing transformation frameworks between these electronic systems.

Keywords: Electronic system, transformation, e-voting, e-auction, e-cash, e-cheque

1 INTRODUCTION

Due to the technological advancement in this digital age, traditional systems have been gradually replaced by electronic systems which are more efficient with the least amount of resources. Cryptographic primitives or protocols play an important role as the underlying building blocks in order to develop secure electronic systems. Traditional systems have their respective security settings, similarly, cryptographic electronic systems are developed in such a way that the required security properties are preserved. In this research, we focus specifically on electronic systems which employ cryptographic primitives as building blocks such as electronic voting (e-voting), electronic auction (e-auction), electronic cash (e-cash), and electronic cheque (e-cheque).

An e-voting system allows the voter to perform vote activity or ballot to make a collaborative decision, to voice out opinion or to vote someone in an election by using e-voting machine in polling stations, or remote voting activity through the internet. According to Jan et al. (2001), Chaum (1981) was the first who introduced cryptographic e-voting system.

An e-auction system was introduced by Franklin and Reiter (1996). An e-auction allows the auctioneer to offer products, commodities or services on an auction site on the internet. Interested individuals may submit their bid for the good in a specific period of time.

An e-cash system was first proposed by Chaum et al. (1982). The e-cash transaction has been implemented in two models which are online e-cash and offline e-cash. Online e-cash require the seller to contact the bank for every single transaction. Verification process in e-cash system is similar with the concept in credit card systems. While in offline e-cash, it allows users to complete a transaction without involving a bank directly. Offline identified e-cash can only be tracked if the e-cash is duplicated and spent.

An e-cheque system was first proposed by Chaum et al. (1988). An e-cheque is the electronic version of a paper cheque. It was designed to perform payment made via network and written by using electronic device.

The fundamental requirements and expectations when designing these systems are the system efficiency, cost-effectiveness, scalability, user-friendliness and most importantly preserving the relevant security properties. Some of these systems share similar set of security requirements. However, to the best of our knowledge, little research has been carried out to provide a detailed comparison analysis between these closely related systems. It is therefore imperative to explore the relationship among these electronic systems by expanding and formalising the transformation frameworks based on the established security notions between them. We manage to summarise the current relationship based on the existing transformation frameworks and provide some research directions.

2 STRUCTURE AND SECURITY PROPERTIES OF ELECTRONIC SYSTEMS

2.1 Structure of e-Voting System (Tso et al., 2019)

An e-voting system consists of three algorithms $\{$*Register, Vote, Open*$\}$, where:

Register: First, the registration server verifies if the voter is a registered voter. Verified voter will obtain a unique virtual identification code (PID). Voter used PID to request unique voting certification ($Cert$) from the registration server.

Vote: Voter uses the $Cert$ to verify their voting qualifications with verification server. Verified voter will receive a ballot signature and personal key pair, then the voter is allowed to cast a vote.

Open: Voter may review ballot information and verify if the certification, ballots information and the results announced by record centre are returned correctly. If there are any conflicts between the ballots information and the announced results, voter can request for ballot verification to ensure the fairness of the election.

2.2 Structure of e-Auction System (McCarthy. et al., 2014)

An e-auction system consists of four algorithms $\{$*Setup, Bid, Open, Reveal*$\}$, where:

Setup: Generate public key pk and private key sk for an auctioneer, ensure that the key is built correctly, and setting up a bulletin board bb.

Bid: Bidder places a price and uses pk of auctioneer to encrypt the price. Bidder then sends the encryption to the auctioneer, the auctioneer will justify the eligibility of bidder and prove that the bidders place their bids once. The auctioneer will publish the bids on the bb once it passed all the checking.

Open: The auctioneer homomorphically combines the ciphertexts which consist of the highest price bid and then decrypts the homomorphic combination. This process repeats on the ciphertext with lower price and will be stopped when the sum of the decrypted text is greater or equal to the amount of items to be sold. The winning bid will be announced.

Reveal: The auctioneer homomorphically combines the ciphertexts of the price that is greater or equal to the winning price, then decrypts the homomorphic combination.

2.3 Structure of e-Cash System (Saputra and Supangkat, 2014)

An e-cash system consists of three algorithms $\{$*Withdrawal, Spending, Deposit*$\}$, where:

Withdrawal: Creation of e-cash. Payers are required to deposit real money to the trusted third party TTP with the same value of e-cash that they requested from the TTP.

Spending: Payer transfers e-cash to the payee. The payee will verify the transaction data with the TTP.

Deposit: Payee deposits the e-cash paid by the payer to the TTP.

2.4 Structure of e-Cheque System (Yeow et al., 2017)

An e-cheque system consists of three algorithms $\{$*Register, Write, Transfer*$\}$, where:

Register: Public parameters $param$ that consist of the secret key sk and public key pk for bank B, payer PR, and payee is generated by the TTP.

Write: PR writes the e-cheque to make the payment to the payee. The PR creates a valid e-cheque containing payment details and hidden PR's account information then submits to B.

Transfer: B first authenticates the e-cheque owner. B will deduct from PR account if there is sufficient money in the account to clear the payment to the payee.

2.5 Security Properties of Electronic Systems

We have reviewed the important security requirements for each electronic system based on the past research (Yeow et al., 2015), (Her et al., 2005), (Magkos et al., 2002), (Yeow et al., 2017) and the detailed comparison analysis between these electronic systems is as shown in Table 1.

Table 1: Security Requirements of Electronic Systems

	e-Voting	e-Auction	e-Cash	e-Cheque
Integrity	Nobody can change the casted vote.	No one is able to change the bidding prices after the bidder placed the bid.	The bank used to prevent the e-cash used by same user for multiple times.	
Non-Repudiation	Voter cannot deny after he had cast a vote.	The bidder cannot deny after he placed a bid.	The bank cannot deny the e-cash transaction.	
Anonymity	The identity of the voter remains anonymous and it is not linkable between the voter and his vote.	The identity of the bidder remains anonymous.	The identity of a user remains anonymous to the merchant.	Only the bank is able to gain access to bidders information.
Authentication	Only eligible voter is allowed to cast vote.	The identity of the bidder and the auctioneer needs to be verified before the bidding process.	The bank needs to verify the payer and payee for the settlement.	
Unforgeability	Voter cannot forge a valid ballot by himself unless he knows the voting authorisation centres secret key.	The bidder or the auctioneer cannot forge a valid bid.	The unauthorised entity cannot forge valid e-cash data.	A valid signed e-cheque of a user should not be mimicked.
Fairness	During the voting phase, voting authorisation centre cannot receive any results that will affect the election.	During bidding and opening phase, the transaction has to be honest to the winning bidder and auctioneer.		
Confidentiality	The voters votes are secret to others.	Only the winning bid is revealed to public and to the auctioneer, other bidding prices remain secret.	The bank only knows the price of the transaction.	Only the bank knows the information of unused and invalid e-cheque.
Privacy	The identity of voter must be secret.	The identity of the bid remains secret except for the winning bid.		
Receipt-Freeness	The voting information cannot be proved to anyone.	The bidding information cannot be proved to anyone.		
Unreuseability	Registered voters are allowed to cast only one time.			
Uncoercibility	All voters cannot sell or prove his vote to information buyer or an adversary.			

3 UNDERLYING TOOLS USED IN ELECTRONIC SYSTEMS WITH ADDITIONAL PROPERTIES

3.1 Linkable Ring Signatures for e-Voting and e-Cash

Tsang and Victor (2005) proposed short linkable ring signatures for e-voting and e-cash where linkable ring signature is a ring signature scheme with added linkability. They claimed that there is no satisfying construction of group/ring signature in e-voting. The first problem showed in past research was that most group signature schemes cannot detect double-voting because the scheme is unlinkable. Second, group signature scheme always possesses the property of anonymity revocation. They proposed a linkable ring signature scheme that provided no anonymity revocation and the scheme is able to detect double-voting and double spenders. One drawback of their proposed scheme is the time delay to effective tagging and small punishment for the offense.

3.2 Deniable Encryption Scheme for e-Voting and e-Auction

According to Howlader et al. (2011), in traditional election and auction system, no receipt is generated for the vote cast/bid place. If the adversary forces the voter/bidder to cast a specific value, the voter/bidder may cast opposite vote or place bid with different value without worrying that the adversary may ask for the proof. However, in e-voting and e-auction, a receipt of vote cast and bid place will be generated. The generated receipt may lead to vote-selling, bid-selling and coercing issues. The deniable encryption scheme still allows an adversary to eavesdrop the communication. It provides fake messages and fake randomness which looks exactly the same with the ciphers. The voter/bidder can give fake data to the adversary without the fear of being caught by the adversary for giving such invalid data. Hence, eavesdropping does not work for the adversary. Besides that, the proposed deniable encryption scheme can be reconstructed to provide deniable mixnet between the user and the authorities. In deniable mixnet scheme, the mixnet servers will share the private key of the deniable encryption. Candidates will send the anonymous encrypted private values to the authorities. The adversary will not be able to find the encrypted vote/bid belongs to whom even the adversary is being provided with the anonymous encrypted vote/bid.

3.3 Uncoercible e-Voting and e-Auction

Burmester et al. (2004) stated that uncoercibility in e-auction and e-voting is important to prevent collusions. The communication channel between the bidder/voter and the bidding/voting authorities must be private, receipt-free and authenticated in order to achieve uncoercibility. The current solutions to solve the uncoercibility in e-voting and e-auction are virtual booths and substituted the untappable channel with tamper-resistant tokens, e.g. smartcards. These proposed schemes provide uncoercibility if the Decision Diffie-Hellman problem is hard.

3.4 Mixnet Model in e-Voting

Her et al. (2005) had introduced several e-voting systems based on the mixnet scheme. There are two methods that used the mixnet in e-voting scheme. In the first method, the voting list is mixed by using mixnet. For an illustration, the mix-centre will mix the voting list and after the mixing, the result will be securely sent to the voter. The voter receives last voting list from the last mix-centre, voter will choose the vote from last voting list. In the second method, the mix-centre mixes the voters encrypted votes. The last mix-centre will decrypt the mixed and encrypted the votes. After that, the last mix-centre will compute the final tally.

3.5 Threshold Party Model in e-Voting and e-Auction

Her et al. (2005), introduced the threshold party model in e-voting by using both El Gamal encryption and (t+1,N) secret sharing scheme. At most t authorities secret value could be revealed. A secret key can be measured by applying the Lagrange interpolation, and in El Gamal decryption from the t+1 known values, the vote can be directly retrieved. Large computing resources are required in the proposed 1-out-of-L voting systems. Magkos et al. (2002) introduced a threshold trust model in e-auction scheme. There are m auctioneers in the threshold trust model, more than m/3 or m/2 are presumed to be trusted. An inefficient technique of secure multi-party function evaluation is used by the auctioneers together to compute the winning price.

3.6 Receipt-Free Scheme in e-Voting and e-Auction

Benaloh and Tuinstra (1994) first proposed the receipt-free scheme for the e-voting system. They used a voting booth to represent the physically secret communication between authorities and voters. Benaloh and Tuinstra (1994) presented two voting protocols that used homomorphic encryption. The first protocol used a single authority and the second protocol used a multi-authority. e-Voting protocol which based on single authority has the weakness. First, during the single authority enforcement of receipt-freeness, it has a weakness in maintaining vote secrecy. Second, single authority knows how each voter casts a vote. The e-voting protocol which used the multi-authority is not receipt-freeness. Her et al. (2005) introduced the receipt-free scheme in case of e-voting where it published the summation of all ballots. The purpose of the receipt-free scheme is to ensure privacy. Therefore, the relation between a voter and a ballot should remain secret. Her et al. (2005) introduced the receipt-free scheme in e-auction where bid-rigging can be prevented by using receipt-free scheme in e-auction. In e-auction, there is a serious issue where a coercer is able to win in all the auctions as the coercer is able to control the winning price if the e-auction did not implement the receipt-free scheme. The coercer is able to win every e-auction with unacceptable low price. In e-auction, it requires the highest price so that it can publish the winning bidder and his winning bid. Due to the last publishing, everyone will know the identity of the winning bidder and his relation with his bidding price.

3.7 On-Line e-Cheque System with Mutual Authentication

Chang et al. (2009) used blind signature, one-way hash function, and RSA digital signature to construct an e-cheque system that fulfils uniqueness, mutual authentication, robustness and non-repudiation. Uniqueness: Payers identity must be attached to the e-cheque so that the bank can easily verify the e-cheque. Mutual Authentication: Both payer and payee can verify each of their identity. Robustness: Only authentic payer and bank can generate e-cheque. Non-Repudiation: The payer cannot disavow after he has signed the e-cheque.

3.8 Blind Signature Scheme in e-Cash

Chaum (1982) proposed the first blind signature e-cash system. The proposed scheme is as follows:

The payer randomly chooses a number x as $r(x)$ and generates $c(x)$ and sends the $c(x)$ to the bank. The bank generates $s'(c(x))$ as it signs on $c(x)$ and withdraws from payers account. The bank sends the $s'(c(x))$ to the payer, payer then generates $c'(s'(c(x))) = s'(x)$ and verifies if $s(s'(x)) = x$. The transaction will be terminated if the result is false. The payer sends $s'(x)$ to payee. The payee will then authenticate the $s'(x)$ by $r(s(s'(x)))$, the transaction will be stopped if the result is false. Payee sends the $s'(x)$ to the bank, the bank will then verify $r(s(s'(x)))$ and the transaction will be stopped if the result is false while if the result is correct, the bank will add it to the database. If the note already exists in the database, the transaction will be terminated. Lastly, the bank will then update the payee of acceptance.

4 REVIEW OF THE EXISTING TRANSFORMATION FRAMEWORKS BETWEEN ELECTRONIC SYSTEMS

4.1 From e-Cash to e-Auction

Choi et al. (2012) proposed an e-auction scheme from e-cash scheme in universal composability (UC) framework. They first noticed alike in the security properties between e-auction and e-cash. By utilising these similarities, they analysed in detail the relationship and the necessity of each problem, and further transformed from e-cash hybrid to e-auction. The following theorem shows the security of the transformed scheme by Choi et al. (2012):

Theorem 4.1. *The auction protocol UC-realises auction functionality as long as at most one of the two authorised agents is semi-honestly corrupted.*

4.2 From e-Cash to e-Voting

Choi et al. (2012) proposed an e-voting scheme from e-cash scheme in UC framework. They first noticed common security properties between e-voting and e-cash. First, the voters vote will be rejected if the voter votes more than once which is similar to double-spending prevention in e-cash. Secondly, voters should be unlinkable to his votes, this is similar in e-cash system where spenders should be unlinkable with the spent e-cash. By utilising these likenesses, they are able to transform an e-cash to e-voting. The following theorem shows the security of the transformed scheme by Choi et al. (2012):

Theorem 4.2. *The vote protocol UC-realises vote functionality against an adversary that compromises the voters and nominees destructively and the authorities semi-honestly, with the constraint of not permitted to compromise the registration authority and nominees within the same period of time.*

Mateu et al. (2014) proposed a transformation of e-cash to e-voting as follows: the bank to authentication server, withdraw protocol in e-cash to voter contact authentication server request a voting credential, e-cash to voting credential that permits the voter to cast vote. They mentioned that the protocol presented can be implemented with a wide range of suitable e-cash systems. An appropriate e-cash system can be selected according to the requirement of e-voting system. The following theorems show the security of the transformed scheme by Mateu et al. (2014):

Theorem 4.3. *If e-cash is untraceable, e-voting guarantees privacy.*

Theorem 4.4. *If e-cash is unforgeable, e-voting provides integrity assuming that the authentication server is honest.*

4.3 From e-Voting to e-Auction

McCarthy. et al. (2014) proposed two transformations of e-auction scheme from e-voting scheme, namely, Hawk e-auction scheme from Helios e-voting scheme and Aucitas e-auction scheme transformed from Civitas e-voting scheme.

Hawk e-auction scheme was derived using a homomorphic encryption scheme that satisfied indistinguishability under chosen-plaintext attack (IND-CPA), proofs of plaintext knowledge, proofs of correct key construction, and proofs of decryption McCarthy. et al. (2014).

Aucitas e-auction scheme, collusion resistance is fulfilled if the bidder is able to persuade a conspirator that they act as stated in the guideline even when the bidder acts differently. In Aucitas, this situation can be fulfilled by prescribed ruling. Bidder creates false credential, bidder uses the fake credential as in the stated rules. In the case that the bidder is given a command to place a bid for a specific value, bidder will create bid by using false credential. In the instruction of Aucitas, this bid will be withdrawn at the verification time of the credential, yet the attacker cannot diagnose this.

Quaglia and Smyth (2018) proposed a Helios family of e-voting system to a secret, verifiable e-auction system. Cryptography primitives that have been used by the latest schemes in e-auction and e-voting are trapdoor bit-commitments, homomorphic encryption, and mixnets. For example, the usage of mixnets in e-voting comes before a similar usage of mixnets in e-auction by more than twenty years. Next, they observed that the security properties between e-auction and e-voting have similarity. The security properties are secrecy and verifiability.

The construction of e-auction scheme Quaglia and Smyth (2018) is based on McCarthy. et al. (2014)'s idea and they improved the work by providing a strong theoretical foundation. The e-auction scheme is constructed from asymmetric encryption scheme which satisfies correctness, completeness, and injectivity. The correctness, completeness, and injectivity of the transformed e-auction scheme Quaglia and Smyth (2018) are based upon similar properties of Helios family of e-voting scheme.

The following theorems show the security of the transformed scheme by Quaglia and Smyth (2018):

Theorem 4.5. *The transformed e-auction scheme satisfied correctness, completeness, and injectivity as an asymmetric encryption scheme with ideal correctness that satisfied indistinguishability under chosen-plaintext attack (IND-CPA).*

Theorem 4.6. *The transformed e-auction scheme satisfied bid secrecy as the asymmetric encryption scheme with perfect correctness satisfied indistinguishability under a parallel chosen-ciphertext attack (IND-PA0).*

Theorem 4.7. *The transformed e-auction scheme satisfied individual verifiability and universal verifiability as Helios e-voting.*

The difference between the work of Quaglia and Smyth (2018) and McCarthy. et al. (2014) is that the former managed to provide security proofs by showing that their derived e-auction scheme satisfies bid secrecy and verifiability. Indeed, Quaglia and Smyth (2018) were the first who introduced the security definitions of e-auction bid secrecy and verifiability.

4.4 From e-Auction to e-Voting

Lipmaa et al. (2003) claimed that the proposed homomorphic scheme of e-auction can serve as a backbone for e-voting system. The proposed homomorphic e-auction scheme by Lipmaa et al. (2003) consists of two phases, namely, bidding phase and bid opening phase that fulfils the security model as follows:

Bipartite threshold trust model: Some of the e-auction functions are run by one set of sellers servers and some are run by the auction authoritys server. The sellers server will need to ensure that the auction authoritys server runs smoothly, and vice versa. Their proposed auction system remains secure if both the sellers server and auction authoritys server are not cheating. They mentioned that this model can be applied in e-voting scheme.

Reduce collusion: Assuming H is a secure commitment system. Normally, some will assume that the H represents hash function. Collusion of auction authority and seller may minimise if the genuine encrypted bids sent to seller after the bidder sends signed bid and commitment to seller and seller then broadcasts all commitments with sellers signature. The e-auction will remain flawless even when auction authority and seller collude. Lipmaa et al. (2003) stated that this simple method is useful in both e-auction and e-voting scheme.

Avoiding replay attacks: In homomorphism, the encoded bid can be generated by everyone. Thus, the cut-and-paste replay attacks can be carried out and compromised the bids privacy. This replay attacks can be prevented by implementing coin-extractability property. Each bid includes a random coin and a unique transaction ID where the random coin was the same as the coin used to encrypt the bid. As long as the random coins are secret to the seller, the replay attacks can be averted. The cut-and-paste replay attacks in homomorphic e-voting scheme could be also prevented with the same mechanism as above.

4.5 From e-Auction to e-Cheque

The e-auction to e-cheque transformation was first proposed by Yeow et al. (2017). They transformed e-auction to e-cheque as follows: bid to cheque, bidder to payer, auctioneer to bank, and one round sealed-bid e-auction (SBEA) acts as a submission of cheque, bulletin board in SBEA to online verification in e-cheque, and payee to a passive party who monitors the transaction between bank and payer to affirm the delivery of e-cheque.

The following theorems show the security of transformed scheme by Yeow et al. (2017). Let SBEA scheme = $\{Setup, Bid, Open\}$ and let e-cheque scheme = $\{Register, Write, Transfer\}$.

Theorem 4.8. *The transformed e-cheque is secure against existential unforgeable under chosen account attack (EUF-CAA) if the underlying SBEA is secure against existential unforgeable under chosen price attack (EUF-CPA).*

Theorem 4.9. *The transformed e-cheque is secure against payer anonymity under chosen account attack (PA-CAA) if the underlying SBEA is secure against bidder anonymity under chosen price attack (BA-CPA).*

Theorem 4.10. *The transformed e-cheque is secure against indistinguishability under chosen cheque attack (IND-CCeA) if the underlying SBEA is secure against indistinguishability under chosen bid attack (IND-CBA).*

4.6 Relationship between the Existing Transformation Frameworks

Based on the above review, Figure 1 summarises the relationship between all existing transformation frameworks of electronic systems. From Figure 1, we can clearly see that e-cash can be transformed to e-voting, e-cash can be transformed to e-auction, and e-auction can be transformed to e-cheque. These three transformations show a one-way transformation. There is an equivalent transformation between e-voting and e-auction, i.e. where e-voting can be transformed to e-auction, and vice versa.

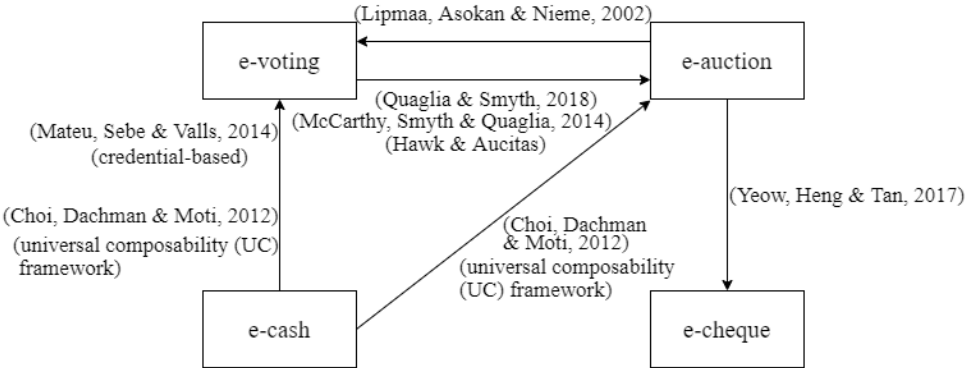

(Lipmaa, Asokan & Nieme, 2002)
e-voting
e-auction
(Quaglia & Smyth, 2018)
(McCarthy, Smyth & Quaglia, 2014)
(Hawk & Aucitas)
(Mateu, Sebe & Valls, 2014)
(credential-based)
(Choi, Dachman & Moti, 2012)
(universal composability (UC) framework)
(Yeow, Heng & Tan, 2017)
(Choi, Dachman & Moti, 2012)
(universal composability (UC) framework)
e-cash
e-cheque

Figure 1: Relationship Diagramme between the Transformation Frameworks

5 POTENTIAL FUTURE WORKS

There remain some open problems from the existing transformation frameworks. Firstly, Choi et al. (2012) constructed e-voting system and e-auction system from e-cash system. In the construction, they did not apply any cryptographic tools or assumptions besides the basic functionality of the systems. They were able to develop a secure e-voting scheme and e-auction system from a given secure e-cash system. The e-voting system is secure against an adversary with the particular corruption pattern. They believed their model is a meaningful starting point.

Secondly, Quaglia and Smyth (2018) constructed an e-auction scheme from an e-voting. Yet, their construction relied on the underlying tally algorithm, a poorly constructed tally algorithm may cause the e-auction scheme to not fulfil bid secrecy. If tally algorithm generates a wrong winning price in the presence of an adversary, a set of bids with the same price will be revealed and the losing bidder will lose his identity secrecy which violates bid secrecy. A tally algorithm can generate correct result under perfect conditions and produce an incorrect result if there exists an adversary. Hence, this causes inconsistency in the result. Moreover, their construction also relied on the reveal algorithm. Reveal algorithm may produce a wrong set of ballots as the output if there is existence of an adversary.

Based on the current review, the following transformations have not been established, namely, transformation from e-voting to e-cash, from e-auction to e-cash, and from e-voting to e-cheque. It would be interesting to carry out further study to see whether these transformations could be established.

5.1 From e-Voting to e-Cash

From Table 1, we can see that both e-voting and e-cash enjoy almost the same security properties, namely, integrity, anonymity, authentication, and unforgeability. With the same security properties, we conjecture that a transformation could possibly be established from e-voting to e-cash. Both systems also share similar in structures with respect to the involved entities. More specifically, voter in e-voting plays a similar role as payer in e-cash system, the authentication server in e-voting plays similar role in e-cash as the bank, and e-voting polling station similar as the role of merchant in e-cash. These descriptions serve as the basic idea of the transformation, an in-depth study could be carried out to see whether this transformation could be established.

5.2 From e-Auction to e-Cash

The basic idea of transformation from e-auction to e-cash can be detailed out as follows: PR contacts B and requests for e-cash. This is done by Setup algorithm in e-auction. TTP acts as B, he first checks if there is sufficient amount to be withdrawn from the PRs account as he requested. After all the checking, the Setup protocol is completed and as a result, the PR is qualified to place a bid. In the spending phase, PR transfers e-cash to payee. This can be done by Bid algorithm in e-auction. Bid plays a role as e-cash, bidder acts as PR, and auctioneer acts as payee. Bidder places bid, then submits to auctioneer. Bidder is allowed to verify the bid placed as the payee verifies the transaction data with a TTP. Payee deposits the e-cash paid by the PR to TTP. This can be done by Open algorithm in e-auction as the auctioneer opens the winning bid. The above discussion serves as the basic idea of potential transformation, a further in-depth study and formal proving are required to be performed.

5.3 From e-Voting to e-Cheque

We have observed that e-voting can be transformed into e-auction and e-auction can be transformed into e-cheque as shown in Figure 1. It is interesting to find out if e-voting can be transformed into e-cheque. e-Voting and e-cheque are having high similarities in their security properties as shown in Table 1, namely, anonymity, unforgeability, and confidentiality. The transformation could be carried out as follows: payer to act as voter, trusted third party to act as authentication server, e-cheque to act as voting credential, and bank to act as polling station. Bulletin board is not required in e-cheque while payee is required in e-cheque. Again, a more formal study is required to confirm our observation.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Malaysia governments Fundamental Research Grant Scheme (FRGS/1/2018/ICT04/MMU/01/01) for supporting this work.

REFERENCES

Benaloh, J. and Tuinstra, D. (1994). Receipt-free secret-ballot elections (extended abstract). In *The 26th Annual ACM Symposium on Theory of Computing*, pages 544553. Association for

Computing Machinery, doi:10.1145/195058.195407.

Burmester, M., Magkos, E., and Chrissikopoulos, V. (2004). Uncoercible e-bidding games. In *Electronic Commerce Research*, volume 4, pages 113–125.

Chang, C., Chang, S., and Lee, J. (2009). An on-line electronic check system with mutual authentication. In *Computers & Electrical Engineering*, volume 35, pages 757–763.

Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. In *Communications of the ACM*, volume 24. pages 8488, Association for Computing Machinery.

Chaum, D. (1982). Blind signatures for untraceable payments. In *Advances in Cryptology*, pages 199–203. Springer US, Springer.

Chaum, D., Fiat, A., and Naor, M. (1988). Untraceable electronic cash. In *Advances in Cryptology — CRYPTO' 88*, volume 403, pages 319–327. Springer New York.

Choi, S., Dachman-Soled, D., and Yung, M. (2012). On the centrality of off-line e-cash to concrete partial information games. In *Security and Cryptography for Networks*, volume 7485, pages 264–280. Lecture Notes in Computer Science.

Franklin, M. and Reiter, M. (1996). The design and implementation of a secure auction service. In *IEEE Transactions on Software Engineering*, volume 22, pages 302–312.

Her, Y., Imamoto, K., and Sakurai, K. (2005). Analysis and comparison of cryptographic techniques in e-voting and e-auction. In *Technical report on Information Science and Electrical Engineering, Kyushu University*, volume 10, pages 91–96. Kyushu University.

Howlader, J., Nair, V., Basu, S., and Mal, A. K. (2011). Uncoercibility in e-voting and e-auctioning mechanisms using deniable encryption. In *International Journal of Network Security & Its Applications*, volume 3, pages 97–109.

Jan, J., Chen, Y., and Lin, Y. (2001). The design of protocol for e-voting on the internet. In *Proceedings IEEE 35th Annual 2001 International Carnahan Conference on Security Technology (Cat. No.01CH37186)*, pages 180–189. doi:10.1109/CCST.2001.962831.

Lipmaa, H., Asokan, N., and Niemi, V. (2003). Secure vickrey auctions without threshold trust. In *Financial Cryptography*, volume 2357, pages 87–101. Springer Berlin Heidelberg.

Magkos, E., Alexandris, N., and Chrissikopoulos, V. (2002). A common security model for conducting e-auctions and e-elections. In *6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers (CSCC 2002)*, pages 7661–7664. http://www.wseas.us/e-library/conferences/crete2002/papers/444-766.pdf.

Mateu, V., Seb, F., and Valls, M. (2014). Constructing credential-based e-voting systems from offline e-coin protocols. In *Journal of Network and Computer Applications*, volume 42, pages 39–44.

McCarthy., A., Smyth, B., and Quaglia, E. (2014). Hawk and aucitas: e-auction schemes from the helios and civitas e-voting schemes. In *Christin N., Safavi-Naini R. (eds) Financial Cryptography and Data Security. FC 2014. Lecture Notes in Computer Science*, volume 8437, pages 51–63. Springer, Berlin, Heidelberg.

Quaglia, E. and Smyth, B. (2018). Secret, verifiable auctions from elections. In *Theoretical Computer Science*, volume 730, pages 44–92.

Saputra, D. and Supangkat, S. (2014). A study of electronic cash paradigm. In *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 273–278. doi:10.1109/ICITSI.2014.7048277.

Tsang, P. and Victor, K. (2005). Short linkable ring signatures for e-voting, e-cash and attestation. In *Information Security Practice and Experience.ISPEC 2005*, volume 3439, pages 48–60, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tso, R., Liu, Z., and Hsiao, J. (2019). Distributed e-voting and e-bidding systems based on smart contract. In *Electronics*, volume 8, page 422.

Yeow, K.-W., Heng, S.-H., and Tan, S.-Y. (2017). From sealed-bid electronic auction to electronic cheque. In *Information Science and Applications 2017*, volume 424, pages 366–376. Springer Singapore.

Yeow, K.-W., Tan, S.-Y., Heng, S.-H., and Behnia, R. (2015). Applications of undeniable signature schemes. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 133–138. doi:10.1109/ICSIPA.2015.7412177.