International Journal of Cryptology Research 5(1): 45-60 (2015)

A New Arbitrated Signing Scheme Based on BFHP

[1]Amir Hamzah Abd Ghafar and [2]Muhammad Rezal Kamel Ariffin
[1,2]Al Kindi Laboratory, Institute for Mathetical Research, Universiti Putra Malaysia, 43400 Serdang ,
[2]Department of Mathematics, Universiti Putra Malaysia, 43400 Serdang
Email: [1]amirghafar87@gmail.com, [2]rezal@upm.edu.my

ABSTRACT

Digital signing is commonly used in any electronic authentication. It preserves data integrity while maintaining non-repudiation value for signer and verifier involved. As digital world strives towards paperless operation, a derivative of digital signing known as arbitrated digital signing fills the need to use a trusted third party (TTP) to monitor the signing and verification process. In this paper, a new arbitrated digital signing scheme based on Bivariate Function Hard Problem (BFHP) is discussed.

Keywords: Bivariate function hard problem, digital signing scheme.

1. INTRODUCTION

Recent computational technologies can handle millions of electronic communications done every day. But for large organizations and corporations which normally handles same amount of communications per second, it can be a tremendous task even when using more advanced technologies. Communications which either happens internally or externally must be authenticated to make sure its digital integrity. Hence, efficient authentication process must be designed.

Authentication involves two parties which is the sender and recipient of the message. As in the real world, sender needs to sign the message to indicate the he is the message's original sender while recipient needs to verify that the signing is true and indeed is the sender's signature. In the digital world involving electronic communications, this is known as the digital signing process.

The notion of digital signing was conjectured by Whittfield (Diffie and Martin Hellman, 1976) in their renowned paper introducing public key cryptography. By using similar concept of public key cryptography, sender signs the message using his or her own private key while recipient will verify by using sender's public key. Several digital signing schemes have been proposed and implemented. Among the extensively used schemes are the RSA (Rivest, Shamir, and Adleman, 1979) and El-Gammal (Gammal, 1985) digital signing schemes. The latter has become a predecessor of Digital Signature Standard (DSS) which is a formal standard endorsed by National Institute of Standards and Technology (NIST) (National Institute of Standards and Technology, 2013).

Both RSA and El-Gammal schemes require a form of modular exponentiation calculation

$$g^a \pmod N$$

for $a \in \mathbb{Z}_N$. The complexity of this calculation if using classical multiplication is $O(n^3)$ if a has same size with N (Galbraith, 2012).. Though this complexity is sufficient to be operated by machines, but for millions of rapid communications, it can be congested. This paper intends to propose a method to reduce the complexity to $O(n^2)$ by using only multiplication and addition operation. Plus, this method is a non-straightforward approach of signing that was proposed by Diffie and Hellman. But it adds the complexity of the scheme to three parties, which fit into some particular scenarios that happened in the mass communication.

The structure of this paper is as follows. We will introduce a concept of hard mathematical problem called bivariate function hard problem in the Section 2. Then, we will give an insight definition of arbitrated digital signing scheme in Section 3 before we propose and discuss our own arbitrated digital signing scheme in Section 4. Then we will end this paper with some future works need to be done and conclude it.

2. PRELIMINARIES

This section intends to brief the hard mathematical problem that acts as the underlying strength for our new scheme.

2.1 Bivariate Function Hard Problem (BFHP)

The following proposition gives a proper analytical description of the Bivariate Function Hard Problem (BFHP).

Definition 2.1. We define $\mathbb{Z}^+_{(2^{m-1},2^m-1)}$ as a set of positive integers in the interval $(2^{m-1}, 2^m - 1)$. In other words, if $x \in \mathbb{Z}^+_{(2^{m-1},2^m-1)}$, x is a m-bit positive integer.

Proposition 2.1. (Ariffin *et al.*, 2013)

Let $F(x_1, x_2, \dots, x_n)$ be a multiplicative one-way function that maps $F\colon \mathbb{Z}^n \to \mathbb{Z}^+_{(2^{m-1}, 2^m - 1)}$. Let F_1 and F_2 be such function (either identical or non-identical) such that $A_1 = F(x_1, x_2, \dots, x_n), A_2 = F(y_1, y_2, \dots, y_n)$ and $\gcd(A_1, A_2) = 1$. Let $u, v \in \mathbb{Z}^+_{(2^{n-1}, 2^n - 1)}$. Let (A_1, A_2) be public parameters and (u, v) be private parameters.

Let

$$G(u, v) = A_1 u + A_2 v \qquad (1)$$

with the domain of the function G is $\mathbb{Z}^2_{(2^{n-1}, 2^n - 1)}$ since the pair of positive integers $(u, v) \in \mathbb{Z}^2_{(2^{n-1}, 2^n - 1)}$ and $\mathbb{Z}^+_{(2^{m+n-1}, 2^{m+n} - 1)}$ is the codomain of G since $A_1 u + A_2 v \in \mathbb{Z}^+_{(2^{m+n-1}, 2^{m+n} - 1)}$.

If at minimum $n - m - 1 = k$, where 2^k is exponentially large for any probabilistic polynomial time (PPT) adversary to sieve through all possible answers, it is infeasible to determine (u, v) over $\mathbb{Z}$ from $G(u, v)$. Furthermore, $(u, v)$is unique for $G(u, v)$ with high probability.

Remark 2.1. We remark that the preferred pair (u, v) in $\mathbb{Z}$, is the *prf*-solution for (1). The preferred pair (u, v) is one of the possible solutions for (1) given by

$$u = u_0 + A_2 t \qquad (2)$$

and

$$v = v_0 - A_1 t \qquad (3)$$

for any $t \in \mathbb{Z}$.

Remark 2.2 Before we proceed with the proof, we remark here that the diophantine equation given by $G(u, v)$ is solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found. That is the BFHP is *prf*-solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found.

Proof. We begin by proving that (u, v) is unique for each $G(u, v)$ with high probability. Let $u_1 \neq u_2$ and $v_1 \neq v_2$ such that

$$A_1 u_1 + A_2 v_1 = A_1 u_2 + A_2 v_2 \qquad (4)$$

We will then have

$$Y = v_2 - v_1 = \frac{A_1(u_1 - u_2)}{A_2}$$

Since $\gcd(A_1, A_2) = 1$ and $A_2 \approx 2^n$, then the probability that Y is an integer is 2^{-n}. Then the probability that $v_1 - v_2$ is an integer solution not equal to zero is 2^{-n}. Thus $v_1 = v_2$ with probability $1 - \frac{1}{2^n}$.

Next we proceed to prove that to *prf*-solved the Diophantine equation given by (1) is infeasible to be solved. The general solution for $G(u, v)$ is given by (2) and (3) for some integer t.

To find u within the stipulated interval $u \in (2^{n-1}, 2^n - 1)$ we have to find the integer t such that the inequality $2^{n-1} < u < 2^n - 1$ holds. This gives

$$\frac{2^{n-1} - u_0}{A_2} < t < \frac{2^n - 1 - u_0}{A_2}$$

Then, the difference between the upper and the lower bound is

$$\frac{2^n - 1 - 2^{n-1}}{A_2} = \frac{2^{n-1} - 1}{A_2} \approx \frac{2^{n-2}}{2^m} = 2^{n-m-2}$$

Since $n - m - 1 = k$ where 2^k is exponentially large for any probabilistic polynomial time (PPT) adversary to sieve through all possible answers, we conclude that the difference is very large and finding the correct t is infeasible. This is also the same scenario for v.■

Example 1. Let $A_1 = 191$ and $A_2 = 229$. Let $u = 41234$ and $v = 52167$. Then $G = 19821937$. Here we take $m = 16$ and $n = 8$. We now construct the parametric solution for this BFHP. The initial points are $u_0 = 118931622$ and $v_0 = -99109685$. The parametric general solution are: $u = u_0 + A_2 t$ and $v = v_0 - A_1 t$. There are approximately $286 \approx 2^9$ (i.e. $\frac{2^{16}}{229}$) values of t to try (i.e. difference between upper and lower bound), while at minimum the value is $t \approx 2^{16}$. In fact, the correct value is $t = 519172 \approx 2^{19}$.

3. ARBITRATED DIGITAL SIGNING

A digital signature scheme which require unconditionally trusted third party (TTP) to become a part of entity who aid the signing and verification process is called arbitrated digital signing scheme (Menezes, Oorschot, and Vanstone, 1997). The TTP may act in roles of an authority body, internal section of a bank or a commercial-based third party. The deal is both Alice and Bob must not have any doubt of information that being sent by TTP. The illustration of the scheme is shown in Figure 1.

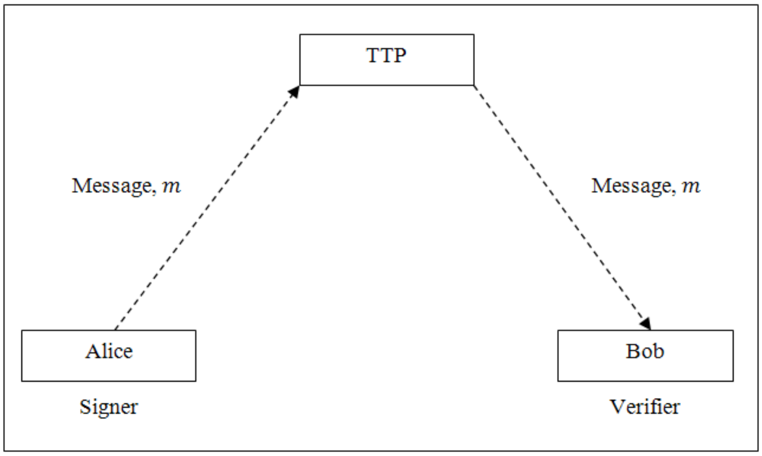

Figure 1: The basic flow of arbitrated signing scheme.

Arbitrated digital signing needs a secure symmetric key encryption. The example of the symmetric key encryption that is used extensively today is AES. The key is used to initiate the communication between TTP, Alice and Bob. This causes a drawback because TTP needs an additional public key communication with entities involved to distribute their secret keys. Our scheme wants to tackle this problem.

We show an arbitrated digital signing scheme with symmetric key encryption in Algorithm 1.

Algorithm 1: Textbook Symmetric Arbitrated Signing Scheme

```
1. Key Generation
    a. Alice and Bob generate their own secret key, $k_A$ and $k_B$
       respectively.
```

   b. Both k_A and k_B are sent to TTP secretly and authentic means to be used as their symmetric key shared with TTP.

2. Signature Generation
   a. Alice calculates message digest of the message, $H = h(m)$.
   b. Alice encrypts H with a symmetric encryption scheme, E using k_A to produce $u = E_{k_A}(H)$.
   c. Alice sends u with her identification string I_A to TTP.
   d. TTP decrypts $E_{k_A}^{-1}(u)$ to get H.
   e. TTP calculates $s = E_{k_T}(H \parallel I_A)$ and sends s to Alice.
   f. Alice's signature is s.
3. Verification
   a. Bob calculates $v = E_{k_B}(s)$.
   b. Bob sends v with his identification string I_B to TTP.
   c. TTP decrypts $E_{k_B}^{-1}(v)$ and get s.
   d. TTP decrypts $E_{k_T}^{-1}(s)$ and get $H \parallel I_A$.
   e. TTP encrypts $w = E_{k_B}(H \parallel I_A)$ and sends w to Bob.
   f. Bob decrypts $E_{k_B}^{-1}(w)$ to get $H \parallel I_A$.
   g. Bob calculates $H' = h(m)$ from m.
   h. Bob accepts Alice's signature if and only if $H' = H$.

The symmetric-key algorithm makes the scheme to be fast. However additional exchanges information between the entities and TTP may cause further risk of being intercepted by the attacker. Hence, we propose a scheme with less additional communication together with an asymmetric encryption scheme which is much faster speed than other commercial public key cryptosystem.

4. NEW ARBITRATED DIGITAL SIGNING SCHEMES BASED ON BFHP

Our new arbitrated digital signing schemes use BFHP as its underlying hard mathematical problem. As mentioned in previous sections, the sizes of both the public and private parameters are vital to ensure the security of the schemes safe.

We refer to two real-world scenarios as basis for our schemes. Both scenarios emphasize on a need for arbitrated characteristics and make our schemes to become scenario-based schemes.

4.1 First Scenario

An operation center of a bank needs details from its clients to complete a financial transaction. However, the center itself is restricted to only trust details that have been verified by another unit or branch from the same or different bank that has a direct contact with clients. This is a real-world scenario occurs in Real-Time Gross Settlement Systems (Bank for International Settlements, 1997) that being used by banks around the world. We propose the scheme in Algorithm 2 that can handle the communications endured in this scenario.

We need to state that the client is in the role of Alice while the trusted unit or branch that have direct contact with clients acts as TTP and the operation center plays Bob's role. We also state here that 2^r is exponentially large.

Algorithm 2: Arbitrated Signing Scheme I

1. Key Generation
   - (a) TTP generates two distinct $r-$bit primes, p_A and p_B.
   - (b) TTP generates two random numbers, d_A and d_T in the size of $r-$bit.
   - (c) TTP computes $e_A \equiv d_A^{-1} (\bmod p_A)$ and $e_T \equiv d_T^{-1} (\bmod p_B)$.
   - (d) Using an asymmetric scheme, TTP encrypts (p_A, d_A) with Alice's public key and (p_B, e_T) with Bob's public key and sends the ciphers to their respective owners.
2. Signature Generation

(a) Alice calculates message digest of the message, $v = h(m)$ in the size of $2r$ – bit.
(b) Alice chooses random secret, u_A in the size of $2r$ – bit.
(c) Alice computes her signature, $S_A = u_A p_A + v d_A$
(d) Alice sends S_A to TTP publicly (without any encryption means).

3. Verification (TTP)
    (a) TTP check whether $S_A = \frac{v}{e_A}$. If yes, reject (i.e It means sender does not sign anything). Else, calculates $W_A \equiv S_A e_A \pmod{p_A}$
    (b) TTP accepts Alice's signature if $W_A \equiv v \pmod{p_A}$.
    (c) TTP chooses random secret key u_T in the size of $2r$ – bit.
    (d) TTP computes $S_T = u_T p_B + v d_T$
    (e) TTP sends S_T to Bob publicly.
4. Verification (Bob)
    (a) Bob check whether $S_T = \frac{v}{e_T}$. If yes, reject (i.e It means sender does not sign anything). Else, calculates $W_T \equiv S_T e_T \pmod{p_B}$.
    (b) Bob accepts Alice's signature if $W_T \equiv v \pmod{p_B}$

Another look at Algorithm 2 is shown in Figure 2.

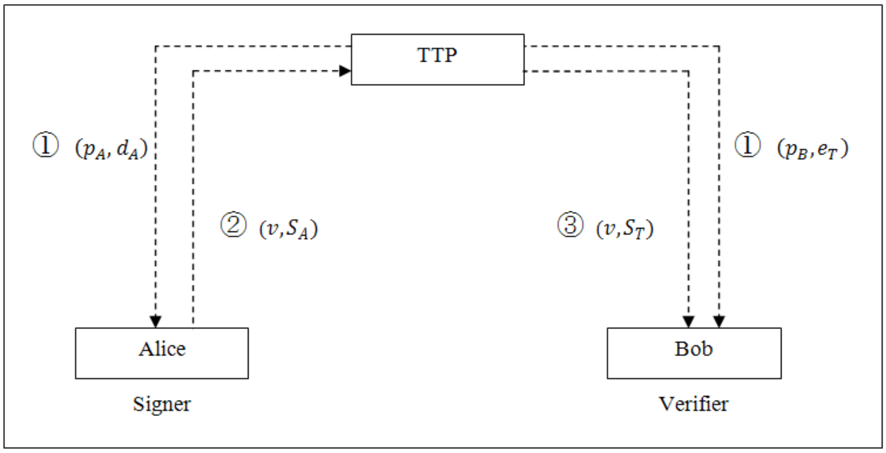

Figure 2: The brief flow of the new arbitrated signing scheme for Scenario 1.

Proposition 4.1 (Completeness) If Alice, Bob and the TTP are honest parties in the proposed arbitrated digital signing scheme in Algorithm 2, Bob will accept Alice's signature.

Proof. If Alice is an honest party, TTP can verify the message using e_A. That is

$$S_A e_A \equiv u_A p_A e_A + v d_A e_A \equiv v \ (\text{mod}\ p_A).$$

Consequently, if TTP is honest, it will sign the verified message from Alice using its private parameter, d_T and Bob can verify the message from Alice is indeed has been verified by TTP before by verifying

$$S_T e_T \equiv u_T p_B e_T + v d_T e_T \equiv v \ (\text{mod}\ p_B).$$

■

4.2 Security Analysis

Remark 4.1. From $S_A = u_A p_A + v d_A$ we can rewrite it as $S_A = X + vY$ where $X = u_A p_A$ and $Y = d_A$ are unknown parameters. Let

$$X = X_0 + vt \qquad (5)$$

$$Y = X_0 - t \qquad (6)$$

be the parametric solution set for S_A. From (5), the interval range for variable t is approximately

$$\frac{2^{3r}}{2^{2r}} \approx 2^r \tag{7}$$

and from (6), it is approximately

$$2^r. \tag{8}$$

Thus, as discussed in section 2, S_A is protected by BFHP.

Proposition 4.2 Given S_A and v, an attacker cannot deduce most significant bits (MSB) of d_A.

Proof. If $\lambda = \frac{u_A p_A}{v}$, we have

$$\frac{S_A}{v} = \lambda + d_A$$

where both $[\lambda]$ and d_A are approximately of the same length. That is $\left\lfloor \frac{S_A}{v} \right\rfloor \neq \lambda$ and $\left\lfloor \frac{S_A}{v} \right\rfloor \neq d_A$. Thus, no information of MSB for d_A will be leaked.

Proposition 4.3 (Forgery Attack) If the attacker is not able to *prf*-solve the BFHP upon S_A and S_T, the proposed arbitrated digital signing scheme in Algorithm 2 can withstand forgery attack.

Proof. S_A is secured by BFHP. That is, the secret key, p_A and ephemeral parameter key, d_A are protected by BFHP. If BFHP can be *prf*-solved, both p_A and d_A can be found. But, based on Proposition 2.1, it is infeasible to find (p_A, d_A). Hence, the proposed scheme can withstand the forgery attack.

The same proof is applied on S_T.

■

Proposition 4.4 (Key only attack) Given e_A, an attacker cannot find the values of secret parameters, (p_A, d_A) if 2^r is exponentially large.

Proof. We can see that $e_A \equiv d_A^{-1} \pmod{p_A}$. Linearly, it can be written as

$$e_A d_A = 1 + \; k_1 p_A$$

for $k_1 \in \mathbb{Z}$. It is trivial to see that attacker will not know value of e_A if he does not have any knowledge of d_A and M_A. However, an attacker can forge a signing by calculating

$$e_A d'_A = 1 + k_2 p'_A$$

for random $d'_A \neq d_A$, $p'_A \neq p_A$ and $k_2 \in \mathbb{Z}$. This will not be the case because during verification, TTP will check if $S'_A = u'_A p'_A + v d'_A$ has been forged by calculating $v (\text{mod}\, p_A)$. It is easy to see that in order for $v (\text{mod}\, p'_A) \neq v (\text{mod}\, p_A)$, the probability is $\frac{1}{2^{4r}}$.

The same proof applies on TTP's signature to Bob.

■

Proposition 4.5 (Known-message attack) Given (S_A, e_A) an attacker that can recompute $v = h(m)$ cannot find the values of secret parameters, (u_A, M_A, d_A).

Proof.

i. The equation $S_A = u_A p_A + v d_A$ consists of 3 variables and protected by BFHP. Thus, (u_A, p_A, d_A) cannot be extracted.
ii. From $e_A \equiv d_A^{-1}\ (mod\, p_A)$, we have $e_A d_A = 1 + k_1 p_A$. This is 1 equiation with 3 variables and protected by BFHP. Thus, (d_A, p_A) cannot be extracted. As a consequence, variable $\mathrm{u_A}$ from $\mathrm{S_A}$ cannot be extracted.

■

Remark 4.2. Proposition 4.5 ensures that no information about (p_A, u_A, k_1) can be obtained. This means the scheme also can withstand chosen-message attack and adaptive chosen-message attack.

Remark 4.3. Proposition 4.5 also shows that every time S_A and $v = h(m)$ changes, the hard problem of BFHP embedded in the scheme still holds. This increases the efficiency of the scheme in rapid communications because Alice does not have to change her secret keys every time she signs a different message to TTP.

4.3 Performance Analysis

To provide the complexity of the scheme, we will use the major operation in our scheme.

Proposition 4.6 For $p_A, p_b \sim 2^r$ bit in size, overall complexity of proposed arbitrated digital signing scheme is $O(r^2)$.

Proof. Both signatures only use multiplication and addition. Hence at most the complexity is $O_1(2r^2)$. The verification involves modular multiplication which at most also produces $O_2(r^2)$.

So, the overall complexity is $O_1(2r^2) + O_2(r^2) = O(3r^2) = O(r^2)$.

■

Now, we propose the second scheme that is based on scenario below.

4.4 Second Scenario

Two entities want to accept an agreement that they have discussed together. However, due to several issues, both do not trust each other. They only trust another third party. In other words, any communication comes from one of the entity will not be trusted by the other. Hence, to indicate both entities have accepted the agreement, they need to sign it and send to the trusted third party (TTP). TTP then will verify both agreements if only if the signed document of the agreement have the same digital fingerprint.

Our next scheme in Algorithm 3 will fit into the environment. The two entities involve in the agreement will be take Alice and Bob's roles and the third party retain TTP's role.

Algorithm 3: AA_β Arbitrated Signing Scheme II

1. Key Generation
   (a) TTP generates two distinct r – bit primes, p_A and p_B.
   (e) Using AA_β encryption scheme, TTP encrypts p_A with Alice's public key and p_B with Bob's public key and sends the ciphers to their respective owners.
2. Signature Generation (Alice)
   (a) Alice calculates message digest of the message, $v = h(m)$ with size of $2r$ – bit.
   (b) Alice chooses random secret keys, d_A in the size of r– bit and U_A in the size of $2r$ – bit.

(c) Alice computes her public parameter $e_A \equiv d_A^{-1} (\bmod\, p_A)$.

(d) Alice computes her signature $S_A = u_A p_A + v d_A$.

(e) Alice sends S_A and e_A to TTP publicly (without any encryption means).

3. Signature Generation (Bob)

   (a) Bob calculates message digest of the message, $v = h(m)$.

   (f) Bob chooses random secret keys, d_B in size of r– bit and U_B in the size of $2r$ – bit.

   (b) Bob computes $e_B \equiv d_B^{-1} (\bmod\, p_B)$

   (c) Bob computes his signature $S_B = u_B p_B + v d_B$.

   (d) Bob sends S_B and e_B to TTP publicly.

4. Verification

   (a) TTP calculates $v = h(m)$.

   (b) TTP calculates $W_A \equiv S_A e_A (\bmod\, p_A)$

   (c) TTP accepts Alice's signature if $W_A \equiv v (\bmod\, p_A)$.

   (d) TTP calculates $W_B \equiv S_B e_B (\bmod\, p_B)$

   (e) TTP accepts Bob's signature if $W_B \equiv v (\bmod\, p_B)$.

Algorithm 3 has the same security and performance features as scheme in Algorithm 2. The only difference is instead of TTP produce a second signature in Algorithm 2, this second scheme requires Bob to sign the document to be verified by TTP. Other than the flow of signature between entities involved, both our proposed scheme has the same structures.

Another look at Algorithm 3 is shown in Figure 3.

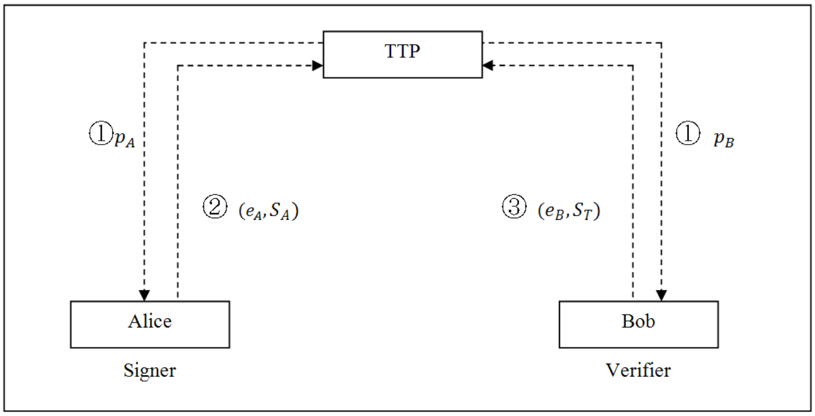

Figure 2: The brief flow of the new arbitrated signing scheme for Scenario 2.

4.5 Comparative Analysis

We show Table 1 to compare our proposed arbitrated digital signing schemes with textbook arbitrated digital signing scheme.

	Textbook Symmetric Arbitrated Digital Signing Scheme (Menezes, Oorschot, and Vanstone, 1997)	New Arbitrated Digital Signing Scheme I	New Arbitrated Digital Signing Scheme II
Major operation	XOR	Multiplication and addition	Multiplication and addition
Number of Signing	4 (stated as encryption)	2	2
Number of Verification	4 (stated as decryption)	2	2
Number of Transmissions	6	4	4

Table 1: Comparative Analysis between Textbook Symmetric Arbitrated Digital Signing Scheme and Proposed Arbitrated Digital Signing Scheme

Although the textbook symmetric arbitrated has a faster XOR operation as major operation, but our proposed arbitrated digital signing schemes have less number in terms of signings, verifications and transmissions have to be done by the entities involved.

5. CONCLUSION

A digital signing scheme must be flexible according to its applications and necessity in real world scenario. We have established two digital signing schemes which involve participation of trusted third party in its signing and verification process. The schemes are scenario-based and do not operate like conventional digital signing schemes. The schemes also use BFHP as its security backbone and we have provided several possible attacks that can be launched onto the schemes. Up to this point, the schemes are still computationally secure and its performance is $O(r^2)$. The schemes also have advantages compared to textbook arbitrated digital signing scheme in terms of number of signing, verification and data transmission between parties involved.

REFERENCES

Ariffin, M., Asbullah, M., Abu, N., and Mahad, Z. 2013. A New Effcient Asymmetric Cryptosystem Based on the Integer Factorization Problem. *Malaysian Journal of Mathematical Sciences*, 7(S):19-37.

Bank for International Settelements. 1997. *Report of Real-Time Gross Settlement Systems.* Retrieved from Bank for International Settlements website: http://www.bis.org/cpmi/publ/d22.pdf.

Diffie, W., and Hellman, M. E. 1976. New Directions in Cryptography. *Information Theory, IEEE Transactions on*, *22*(6), 644-654.

Galbraith, S. D. 2012). *Mathematics of Public Key Cryptography*. Cambridge University Press.

ElGamal, T. 1985, January. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology* (pp. 10-18). Springer Berlin Heidelberg.

Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. 2010. *Handbook of Applied Cryptography*. CRC press.

National Institute of Standards and Technology. 2013. Retrieved from NIST: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

Rivest, R. L., Shamir, A., and Adleman, L. 1978. A Method For Obtaining Digital Signatures And Public-Key Cryptosystems. *Communications of the ACM*, *21*(2), 120-126.