International Journal of Cryptology Research 1(2): 191-204 (2009)

A New Cryptosystem Analogous to LUCELG and Cramer-Shoup

Norliana Muslim and Mohamad Rushdan Md. Said
Institute for Mathematical Research, Universiti Putra Malaysia,
43400 UPM Serdang, Selangor, Malaysia
E-mail: grunge_g5622@yahoo.com, mrushdan@putra.upm.edu.my

ABSTRACT

A special group based on a linear recurrence equation plays an important role in modern cryptography. Its relation appeared differently in various cryptosystem. Some cryptosystems that use this linear recurrence property are LUC, LUCDIF, and LUCELG but the first practical Lucas function in a cryptosystem is LUC, presented by Peter Smith and Michael Lennon in 1993. Cramer-Shoup is a practical public key cryptosystem provably secure against adaptive chosen ciphertext attack that requires a universal one-way hash function. Based on LUCELG and Cramer-Shoup cryptosystems, a new public key cryptosystem is developed by generating the key generation, encryption and decryption algorithm. There are two types of security for the new cryptosystem that we are concerned which are the security of Lucas function and its security against an adaptive chosen ciphertext attack. Since the encryption and decryption algorithm of a new cryptosystem is based on the defined Lucas function, it is believed that the security of Lucas function is polynomial-time equivalent to the generalized discrete logarithm problems. Moreover, the new cryptosystem is secure against adaptive chosen ciphertext attack by assuming that the hash function is chosen from a universal one-way family and the Diffie-Hellman decision problem is hard in the finite field

Keywords: Lucas Function, Public Key Cryptosystem, discrete log problem, adaptive chosen ciphertext attack, hash function.

INTRODUCTION

The ciphers based on the discrete logarithm problem can be implemented using Lucas functions instead of exponentiation. For example, LUC is a public key cryptosystem developed by a group of researchers in Australia and New Zealand (Smith and Lennon, 1993). The Lucas function is a special form of second order linear recurrence relations using a large public integer as modulus (Laih *et al.*, 1994).The cipher implements the analogs of the ElGamal, the Diffie-Hellman key agreement protocol (Diffie and Hellman, 1976), and the RSA system (Rivest *et al.*, 1978) over Lucas sequences (Lucas, 1878). In ElGamal cryptosystem, proposed by Elgamal

(1985), the security relies on the difficulty of computing discrete logarithms. Based on the same difficult mathematical problem as ElGamal, LUCELG uses the calculation of Lucas functions instead of discrete logarithms.

In 1998, Ronald Cramer and Victor Shoup extended ElGamal scheme and developed Cramer-Shoup cryptosystem (Cramer and Shoup, 1998). The scheme and its variants are quite practical and are proved secure against adaptive chosen ciphertext attack under standard intractability assumptions.

In this present paper, we propose a new variation of LUCELG and Cramer-Shoup cryptosystem by showing the key generation, encryption and decryption process. We will continue by discussing the security of the new scheme.

LUCELG CRYPTOSYSTEM

In 1994, Smith (1994), the following cryptographic application of Lucas function an analogue to ElGamal cryptosystem was proposed. The receiver chooses a prime *p* and the initial values P, and $Q = 1$ which are publicized, chosen such that $P^2 - 4Q \mod p$ is a quadratic non-residue, and such that $V_{(p+1)/t}(P,Q) \neq 2 \mod p$, for all $t > 1$ dividing $(p+1)$. Let say Alice sends message to Bob, so Bob (receiver) must choose the private key *x*, and publish the public key $y \equiv V_x(P,Q) \bmod p$.

A message *m* is an integer satisfying $1 \leq m \leq p-1$. To encrypt a message, Alice needs to choose a secret number *k*, which is an integer satisfying $1 \leq k \leq p-1$, calculates $G \equiv V_k(y,O) \bmod p, e_1 \equiv V_k(P,Q) \bmod p$ and $e_2 \equiv Gm \bmod p$. The encrypted message is the pair (e_1, e_2) .

To decrypt the message, Bob needs to compute $V_x(e_1,Q) \equiv V_x(V_k(P,Q),O^k) \equiv V_{kx}(P,Q) \equiv G \bmod p$ and the inverse of *G*. Then Bob can find the message *m*, because $m \equiv e_2 G^{-1} \mod p$.

It is very important that Q is chosen so that $Q \equiv 1 \bmod p$; the recipient needs to know $Q^k \mod p$ for the secret value *k* in order to compute

$V_{kx}\left(P,Q\right)$ from $V_k\left(P,Q\right)$using $V_{kx}\left(P,Q\right)=V_k\left(V_x\left(P,Q\right),Q^k\right)$. This problem can be solved by taking $Q\equiv 1 \bmod p$.

Let $\alpha=\frac{1}{2}\left[P+\sqrt{P^2-4Q}\right]$, and $\Delta=P^2-4Q$; Legendre symbol $\left(\Delta/p\right)=-1$, then $O_{\Delta/P}\in F_p{}^2$,the finite field of p^2element, via an isomorphism that we denote by φ_p. The condition $\left(\Delta/p\right)=-1$is to make sure that one is working in the finite field $F_P{}^2$rather than F .

CRAMER-SHOUP CRYPTOSYSTEM

According to Cramer and Shoup (1998), we assume that we have a group G of prime order q, the plaintext are elements of G and use a universal one-way family of hash functions that map long bit strings to elements of Z_q.

The receiver, Bob pick $g_1,g_2\in G$, $x_1,x_2,y_1,y_2,z\in Z_q$ and compute $c=g_1{}^{x_1}g_2{}^{x_2}$, $d=g_1{}^{y_1}g_2{}^{y_2}$, $h=g_1{}^{z}$. Next, a hash function H is chosen from the family of universal one-way hash functions. The public key is $(g_1,g_2,c,d,h,H)$and the private key is (x_1,x_2,y_1,y_2,z).

Alice as the sender, chooses $k\in Z_q$ and calculates $u_1=g_1{}^k,u_2=g_2{}^k$, $e=h^k m$, $\alpha=H(u_1,u_2,e)$ and $v=c^k d^{k\alpha}$. The ciphertext is $\left(u_1,u_2,e,v\right)$.

Before recover the message, Bob computes $\alpha=H(u_1,u_2,e)$, and tests if $u_1{}^{x_1+y_1\alpha}u_2{}^{x_2+y_2\alpha}=v$. If this condition does not hold, the decryption output is ‘reject’; otherwise, it outputs $m={e}/{u_1^z}$.

We first verify that this is an encryption scheme, in the sense that the decryption of an encryption of a message yields the message. Since $u_1=g_1^k$and $u_2=g_2^k$, we have $u_1^{x_1}u_2^{x_2}=g_1^{kx_1}g_2^{kx_2}=c^k$. Likewise,

$u_1^{y_1}u_2^{y_2} = d^k$ and $u_1^z = h^k$. Therefore, the test performed by the decryption algorithm will pass, and the output will be ${e}/{h^k} = m$.

A NEW CRYPTOSYSTEM ANALOGOUS TO LUCELG AND CRAMER-SHOUP

A new public key cryptosystem using the combination of LUCELG and Cramer-Shoup is proposed as follows.

The receiver chooses a prime p, the initial values P_1, P_2 and $Q = 1$. Let say Alice is the sender and Bob is the receiver, so Bob must choose the private key $(x_1, x_2, y_1, y_2, z) \in F_{p^2}^*$ and compute $c \equiv V_{x_1}(P_1,1) \cdot V_{x_2}(P_2,1) \mod p$, $d \equiv V_{y_1}(P_1,1) \cdot V_{y_2}(P_2,1) \mod p$ and $h \equiv V_z(P_1,1) \mod p$. Here, Bob's public key is $(P_1, P_2, c, d, h, F_{p^2}^*)$ and his secret key is (x_1, x_2, y_1, y_2, z).

To encrypt a message, Alice must represent her message m as an integer such that $1 \le m \le p-1$, choose a secret k such that $1 \le k \le p-1$ and calculates

$$\begin{aligned} &u_1 \equiv V_k(P_1,1) \mod p,\ u_2 \equiv V_k(P_2,1) \mod p, \\ &G \equiv V_k(h,1) \mod p \equiv V_k(V_z(P_1,1),1) \mod p,\ e \equiv Gm \mod p, \\ &\alpha \equiv H(u_1, u_2, e),\ v \equiv V_k(c,1) \cdot V_{k\alpha}(d,1) \mod p \end{aligned}$$

Then, Alice sends the ciphertext (u_1, u_2, e, v) to Bob.

To decrypt the message, Bob use his private key to compute

$$\begin{aligned} m &\equiv \frac{e}{V_z(u_1,1)} \mod p \equiv \frac{Gm}{V_z(u_1,1)} \mod p \\ &\equiv \frac{V_k(V_z(P_1,1),1)}{V_z(V_k(P_1,1),1)} \cdot m \mod p \end{aligned}$$

Example of the New Cryptosystem

Suppose that Bob choose a prime $p=7, P_1=2, P_2=3, Q=1$ and secret key $(2,3,3,2,3)\in F_{7^2}^*$. Later, he calculates

$$
\begin{aligned}
c &\equiv V_{x_1}(P_1,1)\cdot V_{x_2}(P_2,1) \mod p\\
&\equiv V_2(2,1)\cdot V_3(3,1) \mod 7\\
&\equiv 2\cdot 4 \mod 7 \equiv 1 \mod 7\\
d &\equiv V_{y_1}(P_1,1)\cdot V_{y_2}(P_2,1) \mod p\\
&\equiv V_3(2,1)\cdot V_2(3,1) \mod 7\\
&\equiv 2\cdot 7 \mod 7 \equiv 0 \mod 7
\end{aligned}
$$

$$
\begin{aligned}
h &\equiv V_z(P_1,1) \mod p\\
&\equiv V_3(2,1) \mod 7\\
&\equiv 2 \mod 7
\end{aligned}
$$

Bob's public key is $(2,3,1,0,2,F_{7^2}^*)$ and his private key is $(2,3,3,2,3)$.

Let say Alice want to encrypt her message $m=5$ and she choose a secret $k=2$. Then she computes

$$
\begin{aligned}
u_1 &\equiv V_k(P_1,1) \mod p & u_2 &\equiv V_k(P_2,1) \mod p\\
&\equiv V_2(2,1) \mod 7 & &\equiv V_2(3,1) \mod 7\\
&\equiv 2 \mod 7 & &\equiv 0 \mod 7
\end{aligned}
$$

$$
\begin{aligned}
G &\equiv V_k(h,1) \mod p & e &\equiv Gm \mod p\\
&\equiv V_2(2,1) \mod 7 & &\equiv 2\cdot 5 \mod 7\\
&\equiv 2 \mod 7 & &\equiv 3 \mod 7
\end{aligned}
$$

$$
\begin{aligned}
& & v &\equiv V_k(c,1)\cdot V_{k\alpha}(d,1) \mod p\\
\alpha &\equiv H(u_1,u_2,e)\equiv 2 & &\equiv V_2(1,1)\cdot V_4(0,1) \mod 7\\
& & &\equiv 6\cdot 2 \mod 7 \equiv 5 \mod 7
\end{aligned}
$$

Alice sends her ciphertext $(2,0,3,5)$ to Bob.

To recover the message, Bob use his private key to compute

$$m \equiv e \cdot V_z(u_1,1)^{-1} \bmod p \equiv 3 \cdot V_3(2,1)^{-1} \bmod 7$$
$$\equiv 3 \cdot 2^{-1} \bmod 7 \equiv 3 \cdot 4 \bmod 7 \equiv 5 \bmod 7$$

THE SECURITY OF LUCAS FUNCTIONS

The Lucas function is a second order linear recurrence relation which has the form

$$V_n(m) = mV_{n-1}(m) - V_{n-2}(m)\,,\ n \geq 2$$

with an initial value $V_0(m) = 2$, $V_1(m) = m$ where m is an element of multiplicative group of prime number, $F_{p^2}^*$.

The single Lucas problem

Let $m \in F_{p^2}^*$ and z as the Lucas sequence generated by *m*. The single Lucas problem is based on finding an integer *x* such that $V_x(m) = z$.

The double Lucas problem

Let $m_1, m_2 \in F_{p^2}^*$ and z as the Lucas sequence generated by m_1 and m_2 . The double Lucas problem is based on determining the integers x_1 and x_2 such that $V_{x_1}(m_1) \cdot V_{x_2}(m_2) = z$.

All of the Lucas functions and its problem can be seen as a generalization of the exponentiation function but the relationship are not really known. In addition, Laih, Tu and Tai have proposed a theorem by saying; if an algorithm can be used to solve the Lucas problem then it can be used to solve the discrete logarithm problem in polynomial time and also the reverse way (Laih et al., 1995). We proposed the following lemmas and theorems to show that the security of Lucas function is polynomial-time equivalent to the generalized discrete logarithm problems.

Lemma 1. If $g,m\in F_p^*$ and $y=g^x$, then a Lucas sequence $\langle V_t(m)\rangle$ can be construct such that $y+y^{-1}=V_x(m)\in\langle V_t(m)\rangle$ in polynomial time.

Proof: By letting $m=g+g^{-1}$, it is clear that $m\in F_p^*$. Hence, a polynomial $f(x)=x^2-mx+1$ can be construct with *g* and g^{-1} are the two zeroes of *f* (*x*) in F_p^*. Now we let $\langle V_t(m)\rangle$ be the Lucas sequence generated by *m* whose characteristic polynomial is *f* (*x*). Then, $V_t(m)=g^t+\left(g^{-1}\right)^t$. Since $y=g^x$, it yields $y+y^{-1}=g^x+g^{-x}=V_x(m)\in\langle V_t(m)\rangle$. □

Lemma 2. If $m\in F_p^*$, $\langle V_t(m)\rangle$ is the Lucas sequence generated by *m* and $z=V_x(m)\in\langle V_t(m)\rangle$ for some integer *x*, then we can find two elements *g* and *y* in F_p^* such that $z=V_x(m)=y+y^{-1}$ and $y=g^x$ in polynomial time.

Proof: Let $\langle V_t(z)\rangle$ be the Lucas sequence generated by *z*. Then the characteristic polynomial of $\langle V_t(m)\rangle$ is $f^{'}(x)=x^2-zx+1$. Thus, we can find a root *y* of $f^{'}(x)=0$ in polynomial time, where *y* is in F_p^* or $F_{p^2}^*$ which depend upon whether $\left(\dfrac{z^2-4}{p}\right)=1$ or not. If *y* is a root of $f^{'}(x)=0$, then $y+y^{-1}=z$. Like wise, there is an element *g* in F_p^* or $F_{p^2}^*$ in polynomial time such that *g* is a root of characteristic polynomial $f(x)=x^2-mx+1$ for the Lucas sequence $\langle V_t(m)\rangle$. Since $z=V_x(m)\in\langle V_t(m)\rangle$, it gives $z=g^x+g^{-x}$. Combining these two equations of *z* yields, $z=g^x+g^{-x}=y+y^{-1}$.

Theorem 1. If an algorithm *A* can be used to solve the Lucas problem over F_p^*, then *A* can be used to solve the discrete logarithm problem over F_p^* in polynomial time.

Proof: We need to prove that for any given $g,y\in F_p^*$ with $y=g^x$ for some unknown integer *x*, an algorithm *A* can produce *x* as an output in polynomial time.

By using lemma 1, we can find $m \in F_p^*$ and construct the Lucas sequence $\langle V_t(m) \rangle$ such that $y + y^{-1} = V_x(m) \in \langle V_t(m) \rangle$ in polynomial time. Since algorithm A can be used to solve the Lucas problem, it is also can be use to produce the output x such that $V_x(m) = y + y^{-1} = g^x + \left(g^{-1}\right)^x$. □

Theorem 2. If an algorithm $A^{'}$ can be used to solve the discrete logarithm problem over F_p^* or $F_{p^2}^*$, for any $m \in F_p^*$ and $z \in \langle V_t(m) \rangle$, then algorithm $A^{'}$ can be used to produce an integer x such that $V_x(m) = z$ over F_p^* in polynomial time.

Proof: By using lemma 2, we can find g and y in F_p^* or $F_{p^2}^*$ for given $m \in F_p^*$ and $z \in \langle V_t(m) \rangle$ in polynomial time depend upon whether $\left(\frac{z^2-4}{p}\right) = \left(\frac{m^2-4}{p}\right) = 1$ or not such that $m = g + g^{-1}$ and $z = y + y^{-1} = g^x + \left(g^{-1}\right)^x$. If g and y are the input of the algorithm $A^{'}$, then we can obtain output x such that $y = g^x$. Therefore, x is the solution of $V_x(m) = z$. □

From Theorem 1 and Theorem 2, we conclude that the security of Lucas function is polynomial-time equivalent to the generalized discrete logarithm problems.

THE SECURITY AGAINST ADAPTIVE CHOSEN CIPHERTEXT ATTACK

This security was first defined by Rackoff and Simon (1991). In adaptive chosen ciphertext attack, an attacker Eve plays the following game with Alice. First, Alice generates a key with a security parameter as input. Next, Eve makes arbitrary calls to a decryption oracle, providing her own ciphertext and receiving their decryptions. Eve continues by selecting m_0 or m_1 from Alice's messages and sends these to an encryption oracle. The encryption oracle will reply by choosing randomly a bit $b \in \{0,1\}$ and encrypts m_b. Later, Eve again makes calls to the decryption oracle, subject only to the restriction that she may not submit Alice's ciphertext to the oracle verbatim. Finally, Eve outputs $b^{'} \in \{0,1\}$ will be her guess of value b. Eve's advantage is determined to be ε where the probability of her selecting correctly which means $b^{'} = b$ is $\frac{1}{2} + \varepsilon$.

The cryptosystem is said to be secure against adaptive chosen ciphertext attack if no polynomial time algorithm can play these game with advantage a non-negligible function. To prove this, we need to build a simulation that takes as input a quadruple (P_1, P_2, u_1, u_2) coming from either distribution *R* or *D*. Those mean either $(u_1, u_2) = (V_k(P_1, 1), V_k(P_2, 1))$ for some *k* or u_1, u_2 are random elements in $F^*_{p^2}$. The simulation will use this input to role the part of Alice described in the definition of adaptive chosen ciphertext attack. The key generation and an encryption of one of two messages will be done by the simulation. Eve will have access to a working decryption oracle. We must show that if the input comes from the distribution *D*, the simulation is distinguishable from an actual attack. Whenever Eve can correctly distinguish ciphertext, it will succeed in the simulation with the same probability as it would in actual attack.

If the input comes from the distribution *R* then the encrypted message will be perfectly hidden and Eve will not be able to distinguish one ciphertext from another. We will have produced an algorithm that can solve the Diffie-Hellman decision problem with non-negligible problem by declaring that the input came from distribution *D* if Eve guesses correctly and *R* if it does not. If there is no such algorithm exists to break the Diffie-Hellman problem then no algorithm exists to break the new cryptosystem under adaptive chosen ciphertext attack with non-negligible probability.

Theorem 3. The new cryptosystem is secure against adaptive chosen ciphertext attack assuming that:

(1) the hash function *H* is chosen from a universal one-way family
(2) the Diffie-Hellman decision problem is hard in the group *G*.

Proof: The simulator takes as input (P_1, P_2, u_1, u_2). Later, it will play the part of Alice by running the key generation of $(x_1, x_2, y_1, y_2, z_1, z_2) \in F^*_{p^2}$,

$c \equiv V_{x_1}(P_1,1)\cdot V_{x_2}(P_2,1)$ $d \equiv V_{y_1}(P_1,1)\cdot V_{y_2}(P_2,1)$, $h \equiv V_{z_1}(P_1,1)\cdot V_{z_2}(P_2,1)$. The public key is $(P_1,P_2,c,d,h,F^*_{p^2})$ and the private key is $(x_1,x_2,y_1,y_2,z_1,z_2)$. Next, the simulator emulates the decryption oracle for Eve, except outputting $m \equiv \dfrac{e}{V_{z_1}(u_1,1)\cdot V_{z_2}(u_2,1)}$ to account for z_2. The simulator will be given two messages, m_0 or m_1 and randomly selects $b \in \{0,1\}$ and calculates an encryption of

$$e \equiv V_k((V_{z_1}(P_1,1)\cdot V_{z_2}(P_2,1)),1)\cdot m_b,\quad \alpha \equiv H(u_1,u_2,e),$$

and

$$v \equiv V_k\left(V_{x_1}(P_1,1)\cdot V_{x_2}(P_2,1)\right)\cdot V_{k\alpha}\left(V_{y_1}(P_1,1)\cdot V_{y_2}(P_2,1)\right).$$

Thus, the outputs (u_1,u_2,e,v) completed the description of the simulator. When the input is taken from *D*, the simulator's encryption will be a valid ciphertext. Conversely, if the input comes from *R*, the encryption will not be valid because $\log_{P_1} u_1 \neq \log_{P_2} u_2$. □

The following lemma 3 and lemma 4 will continue prove theorem 3.

Lemma 3. When the simulator's input is taken from *D*, the encryption and calls to the decryption oracle are indistinguishable to the adversary from an actual adaptive chosen ciphertext attack.

Proof: The input comes from *D* so $u_1 = V_k(P_1,1)$ and $u_2 = V_k(P_2,1)$. Since u_1,u_2 are indistinguishable from an actual ciphertext, then $e \equiv V_k(h,1)\cdot m$ making e indistinguishable as well. After that, v equation must be verified and Eve can not distinguish the simulation's encryption from a real encryption.

Now, consider two possible types of input $(u_1^{'},u_2^{'},e^{'},v^{'})$ to the decryption oracle. A valid ciphertext happened when $\log_{P_1} u_1^{'} = \log_{P_2} u_2^{'}$.

That means, $u_1^{'} = V_{k^{'}}(P_1,1)$, $u_2^{'} = V_{k^{'}}(P_2,1)$ and the output is $\dfrac{e}{V_{z_1}(u_1,1)\cdot V_{z_2}(u_2,1)} = \dfrac{e}{V_z(V_k(P_1,1),1)}$. Reversely, we will have an invalid ciphertext as a result. □

Lemma 3 follows from the following claim:

Claim 1: The decryption oracle in both actual attack against the cryptosystem and in the simulator rejects all in valid ciphertext except with negligible probability.

Now to prove this claim we need to consider the distribution $P = (x_1, x_2, y_1, y_2) \in F_{p^2}^{*}$ conditioned on Eve's knowledge. We refer $\log()$ to $\log_{P_1}()$ and let $w = \log(P_2)$. Eve can formulate the following two equations to place restrictions on the distribution:

$$\log(c) = \log(V_{x_1}(P_1,1)\cdot V_{x_2}(P_2,1)) = x_1 + wx_2 \qquad (1)$$

$$\log(d) = \log(V_{y_1}(P_1,1)\cdot V_{y_2}(P_2,1)) = y_1 + wy_2 \qquad (2)$$

The decryption oracle will reject the invalid ciphertext $(u_1^{'}, u_2^{'}, e^{'}, v^{'})$ with $\log_{P_1}(u_1^{'}) = r_1$ and $\log_{P_2}(u_2^{'}) = r_2$, unless the test equation of v can be verified. Since x_1, x_2, y_1, y_2 is chosen randomly and not available to Eve, the chance that 2-dimensional plane, P falls onto any line is negligible. Thus, the decryption oracle rejects all invalid ciphertext except with negligible probability.

Lemma 4. When the simulator's input comes from R, the encrypted message is perfectly hidden, and thus cannot be distinguished by Eve.

Proof: Let $u_1 = V_{k_1}(P_1,1)$ and $u_2 = V_{wk_2}(P_1,1)$. We may assume that $k_1 \neq k_2$, since this happens except with negligible probability. Lemma 4 follows from the following two claims:

Claim 2: If the decryption oracle rejects all invalid ciphertexts, then the encrypted message will be hidden from Eve.

Let consider the point $Q = (z_1, z_2) \in F_p^2$. Eve only can access $h = (V_{z_1}(P_1,1) \cdot V_{z_2}(P_2,1))$ before any calls to the decryption oracle. Thus, Q is a random point on the line:

$$\log(h) = z_1 + wz_2 \tag{3}$$

Continuously, Eve can feed the decryption oracle ciphertexts (u_1', u_2', e', v') to gain information about $u_1'^{z_1} u_2'^{z_2} = V_{k_1' z_1}(P_1,1) \cdot V_{k_2' z_2}(P_2,1)$. If the decryption oracle only decrypts valid ciphertext then Eve obtains the information only when $r_1' = r_2'$ and so $V_{k_1' z_1}(P_1,1) \cdot V_{k_2' z_2}(P_2,1) = V_z(V_k(P_1,1))$, yields only the linearly dependant relation $r' \log(h) = r' z_1 + wr' z_2$, giving no further information about Q. In the simulator's encryption (u_1, u_2, e, v), the information about the message m is contained only within $e = \varepsilon \cdot m_b$. This yields the constraint on ε:

$$\log(\varepsilon) = r_1 z_1 + wr_2 z_2 \tag{4}$$

Equation (3) and (4) are linearly independent, so the distribution in Eve's view is uniform. Moreover, $e = \varepsilon \cdot m_b$ constitutes a perfect one-time pad.

Claim 3: The decryption oracle will reject all invalid ciphertexts except with negligible probability.

Now, consider $P = (x_1, x_2, y_1, y_2) \in F_p^4$. This is a random point on the line L which is the intersection of Equation (1) and (2). Thus, from Eve's knowledge *log*(v) can be calculated. However, we have to consider the following three cases when Eve tries to send an invalid ciphertext (u_1', u_2', e', v') to the decryption oracle.

Case 1: If $(u_1', u_2', e') = (u_1, u_2, e)$. The decryption oracle rejects when $v' \neq v$.

Case 2: If $(u_1', u_2', e') \neq (u_1, u_2, e)$ and $\alpha' \neq \alpha$.The decryption oracle rejects unless P lies on equation (1a) and (1b). That means the oracle will rejects since we have linearly independent equations and H intersects L at a single point, except with the negligible probability that this intersection lies only on *P*.

Case 3: If $(u_1^{'}, u_2^{'}, e^{'}) \neq (u_1, u_2, e)$ and $\alpha^{'} = \alpha$. This occurs with only negligible probability since *H* is defined as a collision resistant hash function.

CONCLUSION

The key generation, encryption and decryption algorithm of a new cryptosystem that is analogous to LUCELG and Cramer-Shoup have been defined. The two types of security in the new scheme provide the strengths of the new cryptosystem since it is based on Lucas problem and it is also secure against the most powerful attack. Further research can be continued to develop another cryptosystem by using the third order linear recurrence relation or evaluating the new cryptosystem based on security of pseudorandom generators.

REFERENCES

Cramer, R., and Shoup,V. 1998. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *CRYPTO'98, LNCS 1462*, pp. 13-25.

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

ElGamal, T. 1985. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transaction on Information Theory,* **31**(1985): 469-472.

Laih, C.S., Tu, F.K. and Tai, W.C. 1994. Remarks on the LUC public key system, *Alectronics Lett.* **30**: 123-124.

Laih, C. S., Tu, F. K. and Tai, W.C. 1995. On the security of the Lucas function, *Information Processing Letters*, **53**:243-247.

Lucas, F.E.A. 1878. Theorie des functions numeriques simplement periodiques, *American Jnl Math.*, **1**: 184-240, 289-321.

Rackoff, C. and Simon, D. 1991. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in CryptologyCrypto'91*, pp. 433-444.

Rivest, L., Shamir, A. and Adleman, L. 1978. A method for obtain digital signatures and public key cryptosystem, *Communications of the ACM*, **21**(2): 120-126.

Smith, P., and Lennon, M. 1993. LUC: A new public key system. *Proceedings, Ninth International Conference on Information Security, IFIP/Sec.*

Smith, P. and C.Skinner, 1994. A public-key cryptosystem and a digital signature systems based on the Lucas function analogue to discrete logarithms. *Pre-proceedings Asia Crypt'*94, pp. 298-306.