University of Technology Chemnitz-Zwickau

Department of Computer Science

Theoretical Computer Science
and Information Security

Technical Report

TR-95-1

Digital signature schemes based on Lucas functions

Patrick Horster · Markus Michels · Holger Petersen

February 1995

Straße der Nationen 62
D-09111 Chemnitz, Germany

Phone: (+49) 371-531-1479
Fax: (+49) 371-531-1810

Digital signature schemes based on Lucas functions*

Patrick Horster · Markus Michels · Holger Petersen
Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau
Straße der Nationen 62, D-09111 Chemnitz, Germany
E-mail: {pho,mmi,hpe}@informatik.tu-chemnitz.de

Abstract

In 1993 Lennon and Smith proposed to use Lucas functions instead of the exponentiation function as a one-way function in cryptographic mechanisms. Recently Smith and Skinner presented an ElGamal signature scheme based on Lucas functions.

In this paper we point out a serious weakness in this approach and present our version of an ElGamal signature scheme based on Lucas functions. Furthermore, we outline how to apply the ideas of the Meta-ElGamal signature scheme to Lucas functions. As a result we get various new signature schemes. In contradiction to a conjecture by Smith and Skinner the security of the schemes isn't increased: It can be proved that a variant of the signature schemes based on Lucas functions can be universally forged iff a related signature scheme in GF(p) can be universally forged. We further outline how the Meta signature scheme can be described in an elliptic curve environment and mention some other possible extensions.

Keywords

Cryptography, digital signatures, Lucas functions, elliptic curves

1 INTRODUCTION

In 1984 ElGamal (ElGamal, 1984) published the first signature scheme based on the discrete logarithm problem. Since then a lot of work has been done to modify and generalize this signature scheme. Very important steps of recent research were the discovery of efficient signature schemes with appendix, e.g. by Schnorr, Nyberg and Rueppel or Harn. All these variants can be embedded into a Meta signature scheme with appendix (Horster, Michels and Petersen, 1994). Other signature schemes giving message recovery, e.g. by Nyberg and Rueppel and Piveteau could be embedded into a Meta-Message recovery scheme

*Appeared in Proc. 1st Int. Conference on Communications and Multimedia Security, Graz, Chapman & Hall, (1995), pp. 178–190.

(Horster, Michels and Petersen, 1994).

Since 1981 it has been examined to use other permutations instead of the exponentiation in cryptography. For example, Müller, W.Nöbauer and R.Nöbauer (1981,1985) suggested to use an RSA-like scheme based on the Dickson polynomial. In 1993 a very similar RSA-like scheme has been proposed by Smith and Lennon (1983) based on Lucas functions, indeed, the Dickson polynomial is identical to the Lucas function $V_n(P, Q)$. Recently Smith and Skinner (1994) proposed an ElGamal digital signature scheme based on Lucas functions.

In this paper we will show how to forge this ElGamal digital signature scheme based on Lucas functions universally, point out the design problem and suggest our version of the ElGamal signature scheme based on Lucas functions. Additionally, we will outline how to apply the ideas of the Meta signature scheme with appendix to get various new signature schemes. In Smith and Skinner (1994) it is conjectured that computing the discrete logarithm can be done using a subexponential algorithm while the analogue with Lucas functions can just be done with an exponential algorithm and thus the used parameters can be chosen of smaller size. According to results due to Laih, Tu and Tai (1995) the discrete logarithm problem over Lucas functions is polynomial time equivalent to the discrete logarithm problem in $GF(p)$. Thus this conjecture is wrong. We can further prove that forging one of the new signature schemes based on Lucas functions universally is polynomial time equivalent to forge a related signature scheme in $GF(p)$ universally . As the evaluation of the Lucas functions is slightly less efficient than computing exponentiations – at least by using straightforward techniques – the new schemes are only of theoretical interest. We further outline how the Meta signature scheme with appendix can be described in an elliptic curve environment and mention some other possible extensions.

2 SOME PROPERTIES OF LUCAS FUNCTIONS

Let P and Q be integers and a and b the roots of the equation

$$x^2 - Px + Q = 0.$$

Then

$$U_n(P,Q) := \frac{a^n - b^n}{a - b} \text{ and } V_n(P,Q) := a^n + b^n.$$

As $U_0(P,Q) = 0$, $U_1(P,Q) = 1$, $V_0(P,Q) = 2$, $V_1(P,Q) = P$ it can be shown that $\{U_n(P,Q)\}$ and $\{V_n(P,Q)\}$ are sequences of integers. Note that $P = a + b$ and $Q = a \cdot b$. We further denote $D(P,Q) := (a-b)^2 = P^2 - 4Q$ and L as the Legendre symbol. There are several relations which hold for $U_n(P,Q)$ and $V_n(P,Q)$, (see Lucas 1878, Lehmer 1930, Williams 1982, Smith and Lennon, 1993 for details). The most important for this paper are:

1. $\forall n \in \mathbf{N} : U_n(P \pmod N, Q \pmod N) \equiv U_n(P,Q) \pmod N$,
2. $\forall n \in \mathbf{N} : V_n(P \pmod N, Q \pmod N) \equiv V_n(P,Q) \pmod N$,
3. $V_{nk}(P,Q) = V_n(V_k(P,Q), Q^k)$,
4. $2V_{n+m}(P,Q) = V_n(P,Q)V_m(P,Q) + D(P,Q)U_n(P,Q)U_m(P,Q)$,
5. $U_n(V_k(P,Q), Q^k) = U_{nk}(P,Q)/U_k(P,Q)$,
6. $V_{2n}(P,Q) = V_n^2(P,Q) - 2Q^n$
7. $V_{2n-1}(P,Q) = V_n(P,Q)V_{n-1}(P,Q) - PQ^{n-1}$

8. $V_{n+1}(P,Q) = PV_n(P,Q) - QV_{n-1}(P,Q)$
9. If p is an odd prime, $p \not| Q$ and $L(D(P,Q),p) = \epsilon$, then $U_{(p-\epsilon)m}(P,Q) \equiv 0 \pmod{p}$ and $V_{(p-\epsilon)m}(P,Q) \equiv 2Q^{m(1-\epsilon)/2} \pmod{p}$.

Using the 3rd and 4th relation we see that

$$V_{2k}(P,Q) = V_2(V_k(P,Q), Q^k),$$
$$2V_{k+k}(P,Q) = V_k^2(P,Q) + D(P,Q)U_k^2(P,Q).$$

Therefore

$$U_k(P,Q)^2 = \Big(2V_2(V_k(P,Q),Q^k) - V_k^2(P,Q)\Big)D(P,Q)^{-1} = (V_k^2(P,Q) - 4Q^k)D(P,Q)^{-1}. \quad (1)$$

As we use the equation $x^2 - \lambda x + 1 = 0 \pmod{p}$ in the following, we have $P = \lambda$ and $Q = 1$. We define $U_l(\lambda) := U_l(\lambda,1)$, $V_l(\lambda) := V_l(\lambda,1)$, $V_r(y) := V_r(y,1)$, $V_s(r) := V_s(r,1)$ and $D(\lambda) := D(\lambda,1)$ for simplicity. Then the equation (1) simplifies to:

$$U_k^2(\lambda) = (V_k^2(\lambda) - 4)D(\lambda)^{-1}. \quad (2)$$

Using the equation (2) the 4th relation further simplifies in $GF(p)$ to

$$\begin{aligned} 2V_{n+m}(\lambda) &\equiv V_n(\lambda)V_m(\lambda) + D(\lambda)U_n(\lambda)U_m(\lambda) \\ &\equiv V_n(\lambda)V_m(\lambda) \pm D(\lambda)\sqrt{(V_n^2(\lambda)-4)D(\lambda)^{-1}}\sqrt{(V_m^2(\lambda)-4)D(\lambda)^{-1}} \\ &\equiv V_n(\lambda)V_m(\lambda) \pm \sqrt{(V_n^2(\lambda)-4)(V_m^2(\lambda)-4)} \pmod{p}. \end{aligned}$$

Definition 1 *The discrete logarithm problem over Lucas functions (LDLP) is defined as the following: Given y and l find an element x such that $V_x(l) = y \pmod{p}$.*

3 AN INSECURE SIGNATURE SCHEME BASED ON LUCAS FUNCTIONS

In the following we review the insecure ElGamal signature scheme using Lucas functions proposed by Smith and Skinner (1994):

A prime p is chosen such that $p+1$ is not smooth and a generator λ such that $V_{(p+1)/t}(\lambda) \not\equiv 2 \pmod{p}$ for all t dividing $p+1$ and $L(\lambda^2-4,p) = -1$. Hence we use the equation $x^2 - \lambda x + 1 \equiv 0 \pmod{p}$ as defining equation for U_n and V_n. If $x \in \mathbf{Z}_p$ is the private key of user Alice then $y := V_x(\lambda) \pmod{p}$ and $y_U := U_x(\lambda) \pmod{p}$ are the two public keys of user Alice.

If Alice wants to sign the message $m \in \mathbf{Z}_p$, she chooses a random number $k \in Z_{p+1}^*$, computes $r := V_k(\lambda) \pmod{p}$, $r_U := U_k(\lambda) \pmod{p}$ and solves the signature equation $m \equiv xr + ks \pmod{p+1}$ for the parameter s. Now the triple (r, r_U, s) is the signature for the message m. It can be verified by checking if

$$2V_m(\lambda) \equiv V_r(y)V_s(r) + D(\lambda)y_U U_r(y) r_U U_s(r) \pmod{p}$$

where $D(\lambda) = \lambda^2 - 4 \pmod{p}$.

Using the mentioned relations it is possible to show that the scheme is correct: Note that there exists an integer $i \geq 0$ such that $xr + ks = m + i(p+1)$.

$$\begin{aligned} 2V_m(\lambda) &\equiv V_m(\lambda)2 + D(\lambda)U_m(\lambda)0 \equiv V_m(\lambda)V_{(p+1)i}(\lambda) + D(\lambda)U_m(\lambda)U_{(p+1)i}(\lambda) \\ &\equiv 2V_{m+i(p+1)}(\lambda) \equiv 2V_{ks+xr}(\lambda) \equiv V_{ks}(\lambda)V_{xr}(\lambda) + D(\lambda)U_{xr}(\lambda)U_{ks}(\lambda) \\ &\equiv V_s(V_k(\lambda))V_r(V_x(\lambda)) + D(\lambda)U_x(\lambda)U_r(V_x(\lambda))U_k(\lambda)U_s(V_k(\lambda)) \\ &\equiv V_s(r)V_r(y) + D(\lambda)y_U U_r(y)r_U U_s(r) \pmod p. \end{aligned}$$

Unfortunately the scheme can be easily universally forged. The attacker Carol just has to know the public parameters: If she wants to get a signature for the message m, she chooses $r, s \in \mathbf{Z}_p$ at random, computes $D(\lambda) := \lambda^2 - 4 \pmod p$ and

$$r_U := (2V_m(\lambda) - V_r(y)V_s(r))(D(\lambda)y_U U_r(y)U_s(r))^{-1} \pmod p.$$

Now (r, r_U, s) are the signature parameters for message m. Obviously the signature verification congruence is satisfied. The design problem of this signature scheme is that the relation between r_U and r is not used during signature verification.

4 A NEW SIGNATURE SCHEME BASED ON LUCAS FUNCTIONS

In our proposal of the ElGamal signature scheme based on Lucas functions the relation between r and r_U will be used. Furthermore, the parameter r_U is eliminated, thus the signature consists only of the parameters r and s.

A prime p is chosen such that $p-1$ is not smooth and a generator λ such that $V_{(p-1)/t}(\lambda) \not\equiv 2 \pmod p$ for all t dividing $p-1$ and $L(\lambda^2 - 4, p) = 1$. We use the same defining function $x^2 - \lambda x + 1 \equiv 0 \pmod p$ as above. $x \in \mathbf{Z}_p$ is the secret key of user Alice and $y := V_x(\lambda) \pmod p$ her corresponding public key. If Alice wants to sign the message $m \in \mathbf{Z}_p$, she chooses a random number $k \in Z^*_{p-1}$, computes $r := V_k(\lambda) \pmod p$ and solves the signature equation

$$m \equiv xr + ks \pmod{p-1}$$

for the parameter s. Now the tuple (r, s) is the signature for the message m. Any verifier can check if

$$V_m^2(\lambda) + V_s^2(r) + V_r^2(y) \equiv V_m(\lambda)V_r(y)V_s(r) + 4 \pmod p.$$

To prove the correctness of the verification equation, we need the following lemma:

Lemma 1

$$2V_m(\lambda) \equiv V_r(y)V_s(r) \pm \sqrt{V_r^2(y) - 4}\sqrt{V_s^2(r) - 4} \pmod p$$

Proof.

$$\begin{aligned} 2V_m(\lambda) \equiv&\ 2V_{xr+ks}(\lambda) \\ \equiv&\ V_{xr}(\lambda)V_{ks}(\lambda) \pm \sqrt{(V_{xr}^2(\lambda) - 4)(V_{ks}^2(\lambda) - 4)} \\ \equiv&\ V_r(y)V_s(r) \pm \sqrt{(V_r^2(y) - 4)(V_s^2(r) - 4)} \pmod p \end{aligned}$$

□

Using lemma 1 it is possible to prove the following theorem:

Theorem 1 *The signature scheme is correct.*

Proof.

$$\begin{array}{rrcl}
 & 2V_m(\lambda) & \equiv & V_r(y)V_s(r) \pm \sqrt{V_r^2(y)-4}\sqrt{V_s^2(r)-4} \\
\Leftrightarrow & 2V_m(\lambda) - V_r(y)V_s(r) & \equiv & \pm\sqrt{V_r^2(y)-4}\sqrt{V_s^2(r)-4} \\
\Leftrightarrow & (2V_m(\lambda) - V_r(y)V_s(r))^2 & \equiv & 16 - 4V_s^2(r) - 4V_r^2(y) + V_r^2(y)V_s^2(r) \\
\Leftrightarrow & 4V_m(\lambda)(V_m(\lambda) - V_r(y)V_s(r)) & \equiv & 16 - 4V_s^2(r) - 4V_r^2(y) \\
\Leftrightarrow & V_m(\lambda)(V_m(\lambda) - V_r(y)V_s(r)) & \equiv & 4 - V_s^2(r) - V_r^2(y) \\
\Leftrightarrow & V_m^2(\lambda) + V_s^2(r) + V_r^2(y) & \equiv & V_m(\lambda)V_r(y)V_s(r) + 4 \pmod{p}
\end{array}$$

□

Another signature scheme can be obtained if $L(\lambda^2 - 4, p) = -1$ and therefore the signature equation is computed modulo $p+1$ instead of modulo $p-1$.

4.1 Security analysis

By Smith and Skinner (1994) it is conjectured that computing the discrete logarithm over GF(p) can be done using a subexponential algorithm while the discrete logarithm over Lucas functions can only be computed with an exponential algorithm, e.g. the Baby-step/Giant-step algorithm (Shanks, 1971), and thus all parameters can be chosen of smaller size. This is not true according to a recent result due to Laih, Tu and Tai (1995). They proved that the discrete logarithm over Lucas functions (LDLP) can be done in polynomial time (dependent on the size of the prime p) if the discrete logarithm problem (DLP) in $GF(p)$ and $GF(p^2)$ is feasible. Furthermore, they showed that if LDLP is solvable the DLP in $GF(p)$ is also feasible. As a result LDLP can be solved in subexponential time.

In the following we investigate the universal forgery and existential forgery of our new signature scheme and compare it with the security of an ElGamal like signature scheme which we introduced first.

An ElGamal-like signature scheme

We show that our scheme is equivalent to the following ElGamal-like signature scheme in $GF(p)$.

The trusted authority chooses a large prime p and a generator $\alpha \in \mathbf{Z}_{p-1}$ with order $p-1$. p and α are public system parameters and authentically known to all users. The signer *Alice* chooses a random secret key $x \in \mathbf{Z}_{p-1}$ and computes her public key $y := \alpha^x \pmod{p}$. These values are constant for all messages to be signed. To sign a message $m \in \mathbf{Z}_{p-1}$ Alice chooses a random number $k \in \mathbf{Z}_{p-1}^*$. She computes $r := \alpha^k \pmod{p}$, $\tilde{r} := r + r^{-1} \pmod{p}$ and solves the congruence

$$m \equiv x\tilde{r} + ks \pmod{p-1} \tag{3}$$

for the parameter s. The triple $(m; r, s)$ is the signed message. It can be verified by computing $\tilde{r} := r + r^{-1} \pmod p$ and checking the congruence

$$\alpha^m \equiv y^{\tilde{r}} r^s \pmod p. \tag{4}$$

Universal forgery

An attacker can universally forge the signature if he discovers an effective algorithm $O_E(\alpha, p, y, m) := (r, s)$ with $\alpha^m \equiv y^{r+r^{-1} \bmod p} r^s \pmod p$, where $y := \alpha^x \pmod p$ is the public key of the victim Alice. We need the following lemmata (see Laih, Tu and Tai (1994,1995) and Murphy (1994)):

Lemma 2 *If g and h are elements of a finite field, then*

$$g + g^{-1} = h + h^{-1} \Leftrightarrow g = h \text{ or } g = h^{-1}$$

Proof. There is an element r such that $h = rg$. If $r = 1$ then $g = h$. If $r \neq 1$ then

$$g + g^{-1} = rg + (rg)^{-1} \Leftrightarrow g^2 + 1 = g^2 r + r^{-1} \Leftrightarrow g^2(1 - r) = r^{-1} - 1 \Leftrightarrow g^2 = r^{-1} \Leftrightarrow r = g^{-2}$$

Therefore $g = h^{-1}$. $\square$

Lemma 3 *If $V_n(\lambda)$ is of order t, α is a root of $x^2 - \lambda x + 1$ and α is of order T in $GF(p)$, then $t = T$.*

To forge an ElGamal signature scheme based on Lucas functions universally the attacker has to find an effective algorithm $O_L(\lambda, p, y_A, m) := (r, s)$ with

$$V_m^2(\lambda) + V_s(r)^2 + V_r^2(y_A) \equiv V_m(\lambda) V_r(y_A) V_s(r) + 4 \pmod p.$$

$y_A := V_x(\lambda) \pmod p$ is the public key of the victim Alice.

Theorem 2 *An attacker can universally forge the ElGamal-like signature scheme over $GF(p)$, that is compute $O_E(\alpha, y, p, m) := (r, s)$ effectively, iff he can universally forge an ElGamal signature based on Lucas functions, that is compute $O_L(\lambda, p, y_A, m) := (r, s)$ effectively.*

Proof. "$\Longrightarrow$": Assume there exists an oracle O_E that computes $O_E(\alpha, p, y, m) := (r, s)$ with non-negligible probability $\mathcal{P}$ effectively. Then we construct an effective algorithm for O_L using a polynomial time transformation.

We know the prime p, the generator λ and $y_A := V_x(\lambda) \pmod p$ with $y_A := \alpha^x + \alpha^{-x} \pmod p$ where α is root of $x^2 - \lambda x + 1 \equiv 0 \pmod p$. As we can compute roots modulo a prime efficiently, it is possible to compute α. α is a generator in $GF(p)$, because it is of the same order as λ according to lemma 3. We can also compute the root of $x^2 - y_A x + 1 \equiv 0 \pmod p$ and get y with $y + y^{-1} \equiv y_A \pmod p$. Therefore $y \equiv \alpha^x \pmod p$ or $y \equiv \alpha^{-x} \pmod p$ using lemma 2. We compute $(r, s) := O_E(\alpha, p, y, m)$ with $\alpha^m \equiv y^{r + r^{-1} \pmod p} r^s \pmod p$. Therefore there exists an element k with $r := \alpha^k \pmod p$ and $m \equiv x \cdot (r + r^{-1} \pmod p) + k \cdot s \pmod{p-1}$. Thus we have

$$V_m^2(\lambda) + V_s(V_k(\lambda))^2 + V_{r+r^{-1} \pmod p}^2(y) \equiv V_m(\lambda) V_{r+r^{-1} \pmod p}(y) V_s(V_k(\lambda)) + 4 \pmod p.$$

As $V_k(\lambda) := \alpha^k + \alpha^{-k} \equiv r + r^{-1} \pmod p$ one of the tuples $(r + r^{-1} \pmod p), s)$ and $(-r - r^{-1} \pmod p)), s)$ is the signature on the message m for the new signature scheme based on Lucas functions. As it can be easily checked which is the correct signature using the verification equation the output of O_L is correct with probability $\mathcal{P}$.

"$\Longleftarrow$": Assume there exists an oracle O_L that computes $O_L(\lambda, p, y_A, m) := (r, s)$ with non-negligible probability $\mathcal{P}$ effectively. We derive an effective algorithm for O_E using a polynomial time transformation.

We know p, α and $y := \alpha^x \pmod p$. Then $\lambda := \alpha + \alpha^{-1} \pmod p$ and $y_A := y + y^{-1} \pmod p$. Note that α is a root of $x^2 - \lambda x + 1 \equiv 0 \pmod p$, λ is of the same order as α according to lemma 3. We use O_L to compute $(r, s) := O_L(\lambda, p, y_A, m)$ with

$$V_m^2(\lambda) + V_s(r)^2 + V_r^2(y_A) \equiv V_m(\lambda)V_r(y_A)V_s(r) + 4 \pmod p.$$

We know that there exists a parameter k with $r \equiv V_k(\lambda) \pmod p$ and $m \equiv rx + ks \pmod{p-1}$. Note that $r \equiv \alpha^k + \alpha^{-k} \pmod p$. We can compute a root r' of the equation $x^2 - rx + 1 \equiv 0 \pmod p$. Then we get $r \equiv r' + r'^{-1} \pmod p$ and thus $r' := \alpha^k \pmod p$ or $r' := \alpha^{-k} \pmod p$ according to lemma 2. Therefore either the tuple (r', s) or (r'^{-1}, s) is the signature on the message m for the ElGamal-like signature scheme Again, it can easily be checked which is the correct one. Therefore the output of O_E is correct with probability $\mathcal{P}$. $\square$

Existential forgery

We can show that an attacker can forge our signature scheme existentially, that is he can find m, r, s such that the verification holds but he can't influence the choice of the message m. The attacker chooses $a \in Z_{p-1}$ and $b \in Z_{p-1}^*$ at random and computes

$$r := V_k(\lambda) = 2^{-1}\left(V_a(\lambda)V_b(y) \pm \sqrt{(V_a^2(\lambda) - 4)(V_b^2(y) - 4)}\right) \pmod p$$

As

$$r := V_k(\lambda) \equiv V_{ms^{-1} - xrs^{-1}}(\lambda)$$
$$\equiv 2^{-1}\left(V_{ms^{-1}}(\lambda)V_{-rs^{-1}}(y) \pm \sqrt{(V_{ms^{-1}}^2(\lambda) - 4)(V_{-rs^{-1}}^2(y) - 4)}\right) \pmod p$$

we get $a \equiv ms^{-1} \pmod{p-1}$ and $b \equiv -rs^{-1} \pmod{p-1}$. Therefore the attacker computes $s := -rb^{-1} \bmod p - 1$ and $m := as \pmod{p-1}$, such that (m, r, s) is a valid signature triple.

To avoid this attack, the message should satisfy a redundancy scheme or a hash value of the message should be signed using a collision free, public known hash function.

4.2 Efficiency analysis

The evaluation of the Lucas function $V_b(P)$ can be done using the following algorithm (Williams 1982):

Let $b = \sum_{i=0}^{t} b_i 2^{t-i}$ be the binary decomposition of b and define $f_0 := b_0$ and $f_{k+1} := 2f_k + b_{k+1}$. It is easy to see that $f_t = b$. As $V_0(P) = 2$ and $V_1(P) = P$ we can get the tuple $(V_{f_{k+1}}(P), V_{f_{k+1}-1}(P))$ using $(V_{f_k}(P), V_{f_k-1}(P))$ by the formula

$$(V_{f_{k+1}}(P), V_{f_{k+1}-1}(P)) := \begin{cases} (V_{f_k}^2(P) - 2, V_{f_k}(P)V_{f_k-1}(P) - P) & \text{if } b_{k+1} = 0, \\ (PV_{f_k}^2(P) - V_{f_k}(P)V_{f_k-1}(P) - P, V_{f_k}^2(P) - 2) & \text{if } b_{k+1} = 1. \end{cases}$$

Using this approach $2 \cdot t + wgt(b)$ multiplications, where $t = \lceil \log_2(b) \rceil$, are needed to evaluate $V_b(P)$.

A more efficient evaluation can be obtained by index substitution in the recurrence above and by applying relation (8) in section 2. This approach has independently been proposed by Postl (1988) and by Yen and Laih (1995). We only need $2 \cdot t$ multiplications in this case. The reccurence can be described by the formular

$$(V_{f_{k+1}}(P), V_{f_{k+1}+1}(P)) := \begin{cases} (V_{f_k}^2(P) - 2, V_{f_k}(P)V_{f_k+1}(P) - P) & \text{if } b_{k+1} = 0, \\ (V_{f_k}(P)V_{f_k+1}(P) - P, V_{f_k+1}^2(P) - 2) & \text{if } b_{k+1} = 1. \end{cases}$$

Additionally, Yen and Laih (1995) described another similar algorithm, which has the same complexity. As a result, the evaluation of the Lucas function is slightly less efficient than computing exponentiations (e.g. we need $t + wgt(b)$ multiplications to compute P^b with the square and multiply algorithm). Thus the signature generation (one evaluation of a Lucas function, one inversion) and signature verification (three evaluations of a Lucas function) of the signature scheme based on Lucas function are slightly less efficient than the signature generation (one exponentiation, one inversion) and signature verification (three exponentiations) in the ElGamal signature scheme over finite fields.

It seems possible to find more efficient algorithms using techniques like windowing (Knuth, 1981) or Luc-chains (Yen and Laih, 1995) to evaluate the Lucas functions.

5 THE META SCHEME WITH APPENDIX BASED ON LUCAS FUNCTIONS

In this section we describe how to generalize our ElGamal signature scheme based on Lucas functions using the ideas of the Meta signature scheme with appendix presented by Horster, Michels and Petersen (1994).

The initialization is the same as described the last section. If Alice wants to sign the message $m \in Z_{p-1}$ then she chooses a random number $k \in Z_{p-1}^*$, computes $r' := V_k(\lambda) \pmod p$ and $r := d(r', m)$ using a suitable function d. Then she solves the signature equation $A \equiv xB + kC \pmod{p-1}$ where the coefficients A, B and C are chosen as suitable general functions e, f, g with arguments m, r and s. Now the tuple (r, s) is the signature for the message m.

The verification depends on the properties of the function d: If it is possible to extract r' using the function d^{-1} with $d^{-1}(r, m) = r'$, then any verifier can check if

$$V_A^2(\lambda) + V_B^2(y) + V_C^2(d^{-1}(r,m)) \equiv V_A(\lambda)V_B(y)V_C(d^{-1}(r,m)) + 4 \pmod p.$$

If otherwise C^{-1} modulo $(p-1)$ exists, then the verifier checks whether

$$r = d\Big(2^{-1}(V_{AC^{-1}}(\lambda)V_{BC^{-1}}(y) \pm \sqrt{(V_{AC^{-1}}^2(\lambda) - 4)(V_{AC^{-1}}^2(y) - 4)}) \pmod p, m\Big).$$

The correctness of the first verification can be easily checked similar to theorem 1. The second is obvious because

$$2r' \equiv 2V_k(\lambda) \equiv 2V_{AC^{-1}-xBC^{-1}}(\lambda)$$

$$\equiv V_{AC^{-1}}(\lambda)V_{BC^{-1}}(y) \pm \sqrt{(V_{AC^{-1}}^2(\lambda)-4)(V_{AC^{-1}}^2(y)-4)} \pmod{p}.$$

The requirements for the choice of the functions e, f and g are (Horster, Michels and Petersen, 1994):

1. The parameter r, s and m should appear either in the coefficients A,B or C.
2. The coefficients A,B or C shouldn't be equal to zero and should be pairwise distinct.
3. If m and s appears in one coefficient then m or s should appear at least in one of the two other.
4. If r and s appears in one coefficient then r or s should appear at least in one of the two other.

They also apply to the equivalent forms of the signature equation, which can be obtained e.g. by multiplication or division with one of the coefficients.

5.1 Security analysis

Universal forgery

It is possible to generalize theorem 2. We fix the notation $B[r|\tilde{r}]$ for substituting the parameter r by $\tilde{r}$ in the coefficient B. Consider $O_{E_{A,B,C,d,d_1}}(\alpha, y, p, m) := (r, s)$ as an oracle with

$$\alpha^{A[r|\tilde{r}]} \equiv y^{B[r|\tilde{r}]}d^{-1}(r,m)^{C[r|\tilde{r}]} \pmod{p}$$

where $\tilde{r} := d_1(d^{-1}(r,m) + d^{-1}(r,m)^{-1} \pmod{p}, m)$ and $O_{L_{A,B,C,d_1}}(\lambda, p, y_A, m) := (r, s)$ as oracle, where

$$V_A^2(\lambda) + V_B^2(y) + V_C^2(d_1^{-1}(r,m)) \equiv V_A(\lambda)V_B(y)V_C(d_1^{-1}(r,m)) + 4 \pmod{p}.$$

Note, that the functions d and d_1 must be invertible in this case.

Theorem 3 *An attacker can universally forge a variant of the Meta signature scheme with appendix in $GF(p)$, that is compute $O_{E_{A,B,C,d,d_1}}(\alpha, y, p, m) := (r, s)$ effectively, iff he can universally forge a variant of the Meta signature scheme with appendix based on Lucas functions that is compute $O_{L_{A,B,C,d_1}}(\lambda, p, y_A, m) := (r, s)$ effectively.*

Proof. "$\Longrightarrow$": Assume there exists an oracle O_E that computes $O_{E_{A,B,C,d,d_1}}(\alpha, p, y, m) := (r, s)$ with non-negligible probability $\mathcal{P}$ effectively. Then we try to construct an effective algorithm for $O_{L_{A,B,C,d_1}}$. We know the prime p, the generator λ and $y_A := V_x(\lambda) \pmod{p}$ with $y_A := \alpha^x + \alpha^{-x} \pmod{p}$ where α is root of $x^2 - \lambda x + 1 \equiv 0 \pmod{p}$. As we can compute roots modulo a prime efficiently, it is possible to compute α, which is of the same order as λ according to lemma 3. We can also compute the root of $x^2 - y_A x + 1 \equiv 0 \pmod{p}$ and get y with $y + y^{-1} \equiv y_A \pmod{p}$. Therefore $y \equiv \alpha^x \pmod{p}$ or $y \equiv \alpha^{-x} \pmod{p}$ according to lemma 2. We compute $(r, s) := O_{E_{A,B,C,d,d_1}}(\alpha, p, y, m)$ with $\alpha^{A[r|\tilde{r}]} \equiv y^{B[r|\tilde{r}]}d^{-1}(r,m)^{C[r|\tilde{r}]} \pmod{p}$ and

$\tilde{r} := d_1(d^{-1}(r,m) + d^{-1}(r,m)^{-1} \pmod p, m)$. Therefore it exists an element k with $r' := d^{-1}(r,m) \equiv \alpha^k \pmod p$ and

$$A[r|\tilde{r}] \equiv B[r|\tilde{r}]x + C[r|\tilde{r}]k \pmod{p-1}.$$

Thus we have

$$\begin{aligned} &V^2_{A[r|\tilde{r}]}(\lambda) + V^2_{B[r|\tilde{r}]}(y) + V^2_{C[r|\tilde{r}]}(V_k(\lambda)) \\ \equiv\ &V_{A[r|\tilde{r}]}(\lambda)V_{B[r|\tilde{r}]}(y)V_{C[r|\tilde{r}]}(V_k(\lambda)) + 4 \pmod p. \end{aligned}$$

As $r'_{new} := V_k(\lambda) := \alpha^k + \alpha^{-k} \equiv r' + r'^{-1} \equiv d^{-1}(r,m) + d^{-1}(r,m)^{-1} \pmod p$ we see that $\tilde{r} \equiv d_1(r'_{new}, m)$. As a result $O_{L_{A,B,C,d_1}}$ outputs the correct values $(\tilde{r}, s)$ with probability $\mathcal{P}$.

"$\Longleftarrow$": Assume there exists an oracle O_L that computes $O_{L_{A,B,C,d_1}}(\lambda, p, y_A, m) := (r,s)$ with non-negligible probability $\mathcal{P}$ effectively and try to construct an effective algorithm for $O_{E_{A,B,C,d,d_1}}$. We know p, α and $y := \alpha^x \pmod p$. We compute $\lambda := \alpha + \alpha^{-1} \pmod p$ and $y_A := y + y^{-1} \pmod p$. Note that α is a root of $x^2 - \lambda x + 1 \equiv 0 \pmod p$, λ is of the same order as α according to lemma 3. We use $O_{L_{A,B,C,d,d_1}}$ to compute $(r,s) := O_{L_{A,B,C,d_1}}(\lambda, p, y_A, m)$ with

$$V^2_A(\lambda) + V^2_B(y) + V^2_C(d_1^{-1}(r,m)) \equiv V_A(\lambda)V_B(y)V_C(d_1^{-1}(r,m)) + 4 \pmod p.$$

We know that there exists a parameter k such that $r' = d^{-1}(r,m) \equiv V_k(\lambda) \pmod p$, $r := d_1(r', m)$ and $A \equiv Bx + Ck \pmod{p-1}$. Note that $r' \equiv \alpha^k + \alpha^{-k} \pmod p$. We can compute a root r'_{new} of the equation $x^2 - r'x + 1 \equiv 0 \pmod p$. We get $r' \equiv r'_{new} + r'^{-1}_{new} \pmod p$ and thus $r'_{new} := \alpha^k \pmod p$ or $r'_{new} := \alpha^{-k} \pmod p$ according to lemma 2. Hence $r_{new} := d(r'_{new}, m)$. As a result, we have the equation $r = d_1(d^{-1}(r_{new}, m) + d^{-1}(r_{new}, m)^{-1} \pmod p, m)$. Therefore the tuple (r_{new}, s) is the correct output of $O_{E_{A,B,C,d,d_1}}$ with probability $\mathcal{P}$. □

Existential forgery

In a similar manner as shown in section 4 we can existentially forge every variant of the Meta signature scheme. Of course, this attack can be avoided by signing a hashvalue of the message or by adding redundancy to the message.

5.2 Efficient variants

Note that in some variants the signature generation and signature verification is more efficient than in the scheme presented in section 4, but all the schemes are slightly less efficient than the corresponding schemes in $GF(p)$.

For illustration, we describe an efficient scheme (choose $d(r', m) := r', A := s, B := m \oplus r, C := 1$) more detailed: The intialization is the same as before. If Alice wants to sign the message $m \in \mathbf{Z}_p$, she chooses a random number $k \in Z^*_{p-1}$, computes $r := V_k(\lambda) \pmod p$ and solves the signature equation

$$s \equiv x(m \oplus r) + k \pmod{p-1}$$

for the parameter s. Now the tuple (r,s) is the signature for the message m. Any verifier can check if

$$V^2_s(\lambda) + V^2_{m \oplus r}(y) + r^2 \equiv V_s(\lambda)V_{m\oplus r}(y)r + 4 \pmod p$$

As a result, we need one (off-line) evaluation of the Lucas function for signature generation but no inversion and two evaluations of the Lucas function for signature verification instead of three in the scheme of section 4.

6 FURTHER EXTENSIONS TO OTHER STRUCTURES

The Meta signature scheme with appendix can be extended to other structures, e.g. elliptic curves. Here we outline the scheme, which is a generalization of the Schnorr signature scheme on elliptic curves described by Miyaji (1992) and some ideas of Nyberg and Rueppel (1994).

If K is a finite field F_q of characteristic $\neq 2, 3$, then an elliptic curve is given by

$$E : y^2 = x^2 + ax + b \; (a, b \in K, 4a^3 + 27b^2 \neq 0)$$

The set of K-rational points on E, denoted $E(K)$, is a finite abelian group, where $E(K) := \{(x, y) \in K^2 | y^2 = x^2 + ax + b\}$ and the composition "+" on E is defined as usual (e.g. see Miyaji (1992)).

Definition 2 *Given an elliptic curve $E(K)$ and a basepoint $\mathcal{P} = (p_1, p_2) \in E(K)$ with (prime) order l, the discrete logarithm problem on the elliptic curve (EDLP) is defined as follows: Given $\mathcal{R} = q \cdot \mathcal{P} = \underbrace{\mathcal{P} + \cdots + \mathcal{P}}_{q}$ and $\mathcal{P} \in E(K)$, compute $q \in Z_l$.*

The elliptic curve $E(K)$ has to be chosen carefully (see Miyaji (1992) for detail). Then the discrete logarithm problem on an elliptic curve can only be computed with exponential time algorithms. There exist two approaches to define the Meta signature scheme, which are briefly described here.

First approach

The user Alice chooses an integer $x \in Z_l$ as her secret key and computes her related public key $\mathcal{Y} := x \cdot \mathcal{P}$. To sign a message m she chooses a random number $k \in Z_l$, computes $\mathcal{R}' = (r_1', r_2') := k \cdot \mathcal{P}$, $r := d(r_1', r_2', m)$ and solves the signature equation $A \equiv x \cdot B + k \cdot C \pmod{l}$, where the coefficients A, B and C are chosen as suitable general functions e, f, g with arguments m, r and s. Now the tuple $(\mathcal{R}', s)$ is the signature for the message m. To reduce the signature size, it is also possible to transmit only the first component r_1' of $\mathcal{R}'$ and one bit to specify the choice of the second component r_2' as described by Menezes, Qu and Vanstone (1995). The full component can be recovered from this information. The verification can be done by computing $r := d(r_1', r_2', m)$ and checking if

$$A \cdot \mathcal{P} = B \cdot \mathcal{Y} + C \cdot \mathcal{R}'.$$

Second approach

Another possibility for signing a message $m \in \mathbf{Z}_p$ is to map it on a elliptic curve point $\mathcal{M}$ using a suitable message mapping function. Then Alice chooses a random number $k \in Z_l$, computes $\mathcal{R}' = (r_1', r_2') := k \cdot \mathcal{P}$, $\mathcal{R} := (r_1, r_2) = d(\mathcal{R}', \mathcal{M})$ and solves the signature equation $A \equiv x \cdot B + k \cdot C \pmod{l}$, where the coefficients A, B and C are chosen as suitable general functions e, f, g with arguments m, r_1, r_2 and s. Now the tuple $(\mathcal{R}, s)$ is the signature on the message m. As above, it's possible to use represent $\mathcal{R}$ by the first component r_1 and one bit to fix the second component r_2.

If the function d^{-1} exists and can be evaluated efficiently, the verification can be done by computing $\mathcal{R}' := d^{-1}(\mathcal{R}, m)$ and checking if

$$A \cdot \mathcal{P} = B \cdot \mathcal{Y} + C \cdot \mathcal{R}'.$$

Otherwise, the verifier computes

$$\mathcal{R}' := (AC^{-1} \pmod l) \cdot \mathcal{P} + (-BC^{-1} \pmod l) \cdot \mathcal{Y}$$

and checks if

$$d(\mathcal{R}', \mathcal{M}) = \mathcal{R}.$$

The requirements for the choice of the functions e, f and g are similar as described in section 5, a detailed security analysis should be done in further work.

7 CONCLUSION

In this paper we showed that the ElGamal signature scheme based on Lucas function proposed by Smith and Skinner suffered from a design problem and presented a new ElGamal signature scheme based on Lucas functions which avoids this weakness. We pointed out that security level of this new scheme and the other new schemes derivated using the ideas of the Meta signature scheme with appendix is the same as a related signature scheme in $GF(p)$. As the evaluation of the Lucas function is slightly less efficient than computing exponentiations, the efficiency of the new schemes is lower than the related signature scheme in $GF(p)$. It is straightforward to derivate signature schemes giving message recovery using the ideas given by Horster, Michels and Petersen (1994).

We have outlined how the Meta signature scheme with appendix can be described in an elliptic curve environment. Further work can be done to extend the scheme to other structures, e.g. using Redei-functions or real quadratic fields.

8 ACKNOWLEDGEMENT

We thank Prof. C.-S. Laih from National Cheng Kung University, Tainan, Taiwan, R.O.C., for sending us the preprint of the paper by Laih, Tu and Tai (1995).

REFERENCES

T.ElGamal, (1984), Cryptography and logarithms over finite fields, *Ph.D. thesis*, Stanford University, CA., UMI Order No. DA 8420519, 119 pages.

P.Horster, M.Michels, H.Petersen, (1994), Meta-ElGamal signature scheme for one message block, *Proc. of the Workshop IT-Security*, Vienna, Sep. 22-23, 1994, R. Oldenbourg Wien München, 1995, pp. 66–81.

P.Horster, M.Michels, H.Petersen, (1994), Meta-ElGamal signature scheme, *Proc. of the 2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia, Nov. 2–4, 1994, pp. 96–107.

P.Horster, M.Michels, H.Petersen, (1994), Meta Message Recovery and Meta blind signature schemes based on the discrete logarithm problem and some applications, Lecture Notes in Computer Science 917, *Advances in Cryptology: Proc. Asiacrypt '94*, Berlin: Springer Verlag, 1995, pp. 224 – 37.

D.E.Knuth, (1981), The art of computer programming, Vol. 2: Seminumerical algorithms, 2nd Edition, Addison-Wesley, Reading, MA.

C.-S.Laih, F.-K.Tu, W.-C.Tai, (1994), Remarks on LUC public key system, *Electronics Letters*, Vol. 30, No. 2, pp. 123–4.

C.-S.Laih, F.-K.Tu, W.-C.Tai, (1995), On the security of Lucas function, *Information Processing Letters*, Vol. 53, pp. 243–7.

F.E.A.Lucas, (1878), Theorie des fonctions numeriques simplement periodiques, *American Journal Mathematics*, Vol. 1, pp. 184–240 and 289–321.

D.H. Lehmer, (1930), An extended theory of Lucas' functions, *Annals of Mathematics* (2), Vol.31, pp. 419–48.

A.J.Menezes, M.Qu, S.A.Vanstone, (1995), Standard for RSA, Diffie-Hellman and related public-key cryptography, Part 6: Elliptic curve systems (Draft 3), Working Draft, IEEE P1363 Standard, January, 42 pages.

A.Miyaji, (1992), Elliptic curves over F_p suitable for cryptosystems , Lecture Notes in Computer Science 718, *Advances in Cryptology: Proc. Asiacrypt '92*, Berlin: Springer Verlag, 1993, pp. 224 – 37.

W.B.Müller, W.Nöbauer, (1981), Some remarks on public key cryptosystems, *Studia Sci. Math. Hung.*, Vol. 16, pp. 71–6.

W.B.Müller, R.Nöbauer, (1985), Cryptanalysis of the Dickson-scheme, Lecture Notes in Computer Science 219, *Advances in Cryptology: Proc. Eurocrypt'85*, Berlin: Springer Verlag, 1986, pp. 50–61.

S.Murphy, (1994), Comment: Remarks on LUC public key system, *Electronics Letters*, Vol. 30, No. 7, pp. 558–9.

K.Nyberg, R.A.Rueppel, (1994), Message recovery for signature schemes based on the discrete logarithm problem, 21 July 1994, to appear in Design, Codes and Cryptography, Kluwer Academic Publishers, Boston, 15 pages.

H.Postl, (1988), Fast evaluation of Dickson Polynomials, *Contributions to General Algebra 6*, Verlag Hölder-Pichler-Tempsky, Wien - Verlag B.G. Teubner, Stuttgart, pp. 223–5.

D.Shanks, (1971), Class number, a theory of factorisation and genera, *Proceedings Symposia in Pure Mathematics* (20), Providence: American Mathematical Society, pp. 415–40.

P.Smith, C.Skinner, (1994), A public key cryptosystem and a digital signature scheme based on Lucas functions analogue to discrete logarithms, Lecture Notes in Computer Science 917, *Advances in Cryptology: Proc. Asiacrypt '94*, Berlin: Springer Verlag, 1995, pp. 357 – 64.

P.Smith, M.Lennon, (1993), LUC: A new public-key system, *Proc. of IFIP/SEC '93*, Elsevier Science Publishers, 1994, pp. 97–110.

H.Williams, (1982), A $p+1$ method of factoring, *Mathematics on Computation*, Vol. 39, pp. 225–34.

S.-M.Yen, C.-S.Laih, (1995), Fast algorithms for the LUC digital signature computation, *IEE Proc.-Comput. Digit. Tech.*, Vol. 142, No. 2, pp. 165–9.