Some Remarks on Lucas-Based Cryptosystems

Daniel Bleichenbacher[1], Wieb Bosma[2], Arjen K. Lenstra[3]

[1] Institut für Theoretische Informatik, ETH Zentrum, 8092 Zürich, Switzerland
E-mail: bleichen@inf.ethz.ch
[2] School of Mathematics and Statistics, University of Sydney,
Sydney, NSW 2006, Australia
E-mail: wieb@maths.su.oz.au
[3] MRE-2Q330, Bellcore, 445 South Street, Morristown, NJ 07960, U.S.A.
E-mail: lenstra@bellcore.com

Abstract. We review the well-known relation between Lucas sequences and exponentiation. This leads to the observation that certain public-key cryptosystems that are based on the use of Lucas sequences have some elementary properties their re-inventors were apparently not aware of. In particular, we present a chosen-message forgery for 'LUC' (cf. [21; 25]), and we show that 'LUCELG' and 'LUCDIF' (cf. [22, 26]) are vulnerable to subexponential time attacks. This proves that various claims that were made about Lucas-based cryptosystems are incorrect.

1 Introduction

The application of Lucas sequences in various branches of number theory is well known (cf. [18]), and their properties have been studied extensively. Applications of Lucas sequences to public-key cryptography, phrased in terms of the equivalent Dickson-polynomials, were proposed and analysed by a series of authors [13; 14; 12; 16; 17; 11]. More recently, the system from [13] re-emerged, by a different author and in slightly altered form, as 'LUC' (cf. [21], and later [25]), and was subsequently extended to 'LUCDIF', 'LUCELG PK', and 'LUCELG DS' (cf. [22; 26]). The difference between [13] and [21; 25] is that the latter introduce 'message-dependent' keys.

The main selling point of the Lucas-based cryptosystems as presented in these later publications (cf. [21; 22; 25; 26]) is that they are not formulated in terms of exponentiation. This would make them unsusceptible to various well-known attacks that threaten the security of more traditional exponentiation-based cryptosystems like 'RSA' (cf. [19]) and 'Diffie-Hellman' (cf. [4]). This is illustrated by the following quotes from [21]:

> This opens RSA to a cryptographic attack known as *adaptive chosen-message forgery*. ... LUC is not multiplicative and therefore not susceptible to this attack.

and from [22]:

> This problem has the advantage that the subexponential algorithms do not appear to generalize to it, so breaking these ciphers is much more expensive.

Concerning the first quote, it was shown independently in [2] and [6] that LUC is susceptible to 'existential forgeries', a restricted variant of chosen-message forgeries. LUC seemed to avoid a true chosen-message forgery, however, which is, according to the response to [6] in [23], 'the most important advance of LUC over RSA'.

Concerning the second quote, LUCDIF and LUCELG would require far shorter key sizes than traditional systems to provide the same level of security. Or, alternatively, with the same key sizes they would provide security far superior to the older systems.

In this paper we address these two quotes. We review the relation between Lucas sequences and exponentiation, and derive some properties of the Lucas-based cryptosystems that the authors of [21; 22; 25; 26] might not have been aware of. As a result, we present a chosen-message forgery for LUC that is more general than the 'existential forgery' referred to above, thus undermining LUC's main advantage over RSA.

Furthermore, we show that LUCDIF and LUCELG are vulnerable to subexponential time attacks[4]. We do not claim that the security of LUCDIF and LUCELG is threatened by these subexponential attacks to the same extent as RSA or standard ElGamal cryptosystems are threatened by subexponential time attacks. In the latter systems one typically works in groups of order $\approx m$, for some integer m. They can be broken in time $L_m[1/3; (64/9)^{1/3} + o(1)]$, for $m \to \infty$, where

$$L_m[u, v] = \exp(v(\log m)^u (\log\log m)^{1-u}),$$

either by factoring m (cf. [10]) or by computing a discrete logarithm in a group of order $\approx m$ (cf. [1; 5; 7; 20]).

The situation for LUCDIF and LUCELG is reminiscent of the Schnorr variation of ElGamal as used in the US government Digital Signature Algorithm ('DSA', cf. [15]). In DSA one works in a subgroup of order q of a group of order $\approx p$, with q substantially smaller than p. As above, DSA can be broken in time $L_p[1/3; (64/9)^{1/3} + o(1)]$, which is subexponential in p, but an attack that is subexponential in the subgroup order q seems to be infeasible. So, in a subexponential attack on DSA nobody knows how to take advantage of the small subgroup size. As we will see below, in LUCDIF and LUCELG one works in a subgroup of order $\approx p$ of a group of order $\approx p^2$. A subexponential attack would require time $L_{p^2}[1/3; (64/9)^{1/3} + o(1)] = L_p[1/3; (128/9)^{1/3} + o(1)]$. Although this is subexponential in p, it is much slower than time $L_p[1/3; (64/9)^{1/3} + o(1)]$ which one would want to take full advantage of the small subgroup size.

[4] This fact was independently noted by Burt Kaliski, Scott Vanstone, and the authors of [9]. We are grateful to an anonymous member of the Crypto'95 program committee for bringing the latter paper to our attention.

This greater resistance against subexponential attacks, however, might be offset by possible greater speed of the more traditional systems, like RSA, if comparable parameter sizes are used. It is conceivable that one could use substantially larger parameters in RSA, and still attain the same speed as a Lucas-based system with smaller parameters. Naturally, this would affect the relative security of the two systems. Because these considerations depend heavily on implementation details, we do not elaborate. In any case, we conclude that the situation is not as bright for LUCDIF and LUCELG as suggested in [26], where it is assumed that the best attacks 'may take time proportional to $p^{1/5}$'.

The paper is organized as follows. First we review some properties of Lucas sequences. Next we present LUC and a chosen-message forgery for LUC, and then we discuss the relative strengths of LUC and RSA. Finally, we present LUCELG PK and a subexponential time attack against it. Similar attacks on LUCELG DS and LUCDIF follow immediately.

2 Lucas sequences

Let P, Q be integers, and let α be a root of $x^2 - Px + Q = 0$ in the field $\mathbf{Q}(\sqrt{\Delta})$, where $\Delta = P^2 - 4Q \in \mathbf{Z}$ is assumed to be a non-square (but not necessarily squarefree). Then α is an element of the ring of integers $\mathcal{O}_\Delta$ of the quadratic field $\mathbf{Q}(\sqrt{\Delta})$, and there exist integers $v = v(\alpha)$ and $u = u(\alpha)$ such that $\alpha = \frac{v+u\sqrt{\Delta}}{2}$. In fact, for every $k \geq 1$ it holds that $2\alpha^k \in \mathbf{Z}[\sqrt{\Delta}]$, and we can write $\alpha^k = \frac{v_k+u_k\sqrt{\Delta}}{2}$, for certain integers $v_k = v(\alpha^k) = v_k(\alpha)$ and $u_k = u(\alpha^k) = u_k(\alpha)$.

Choosing $\alpha = \frac{P+\sqrt{\Delta}}{2}$ and its conjugate $\beta = \bar{\alpha} = \frac{P-\sqrt{\Delta}}{2}$, we find that $v_1(\alpha) = v(\alpha) = P$ and $u_1(\alpha) = u(\alpha) = 1$ and it is easy to see by induction that the v_k and u_k are given by the recurrence relations

$$u_{k+2} = u_{k+2}(P, Q) = Pu_{k+1} - Qu_k, \quad u_1 = 1, \quad u_0 = 0,$$
$$v_{k+2} = v_{k+2}(P, Q) = Pv_{k+1} - Qv_k, \quad v_1 = P, \quad v_0 = 2.$$

Remarks. Thus the v_k, u_k may be seen as the 'coefficients' of the powers of α that may be computed by the above recurrence relations. Knowing v_k and u_k implies knowledge of α^k, which immediately ties the problem of determining k from v_k and u_k to the discrete logarithm of α^k with respect to the base α.

Depending on which view we like to stress we will write $v_k(\alpha)$ or $v_k(P, Q)$, and these are related via $\alpha = \frac{P+\sqrt{P^2-4Q}}{2}$.

Of the many relations between the u_k, v_k we derive a few that are relevant for what is to follow. The first lemma deals with the u and v of conjugates, traces and norms of powers.

Lemma 1. *With notation as above, for every α and every $k \geq 0$:*
(i) $v_k(\beta) = v_k(\alpha)$ *and* $u_k(\beta) = -u_k(\alpha)$.

(ii) $\alpha^k + \beta^k = v_k(\alpha) = v_k(\beta)$.

(iii) $\alpha^k \beta^k = Q^k = \dfrac{v_k^2(\alpha) - \Delta u_k^2(\alpha)}{4}$.

Proof. The first and second assertions are immediate from the fact that exponentiation and conjugation commute:

$$\beta^k = (\bar{\alpha})^k = \overline{\alpha^k} = \frac{v_k(\alpha) - u_k(\alpha)\sqrt{\Delta}}{2}.$$

Multiplying this by α^k yields (iii).

Lemma 2. *For all $k \geq \ell \geq 0$:*

$$v_{k+\ell} = v_k v_\ell - Q^\ell v_{k-\ell}.$$

Proof. Use Lemma 1(ii) and (iii).

This shows that v_k for large k can be easily computed since exponentiation can be done by repeated squaring and multiplication. Alternatively, if both sequences are needed, the following lemma can be used.

Lemma 3. *For $k \geq 1$:*

(i) $u_{2k} = u_k v_k, \quad v_{2k} = v_k^2 - 2Q^k$;

(ii) $u_{2k+1} = (Pu_{2k} + v_{2k})/2$, *and* $v_{2k+1} = (\Delta u_{2k} + Pv_{2k})/2$.

Proof. Write out the coefficients of $(\alpha^k)^2$ and of $\alpha(\alpha^{2k})$ respectively.

The other relevant relation is most easily formulated in terms of recurrent sequences. It expresses the fact that the coefficients of the powers of a fixed power α^m can be found from a recursion with parameters depending on α^m in a simple fashion.

Lemma 4. *For every P and Q:*

$$v_{km}(P,Q) = v_k(v_m(P,Q), Q^m),$$

and

$$u_{km}(P,Q) = u_k(v_m(P,Q), Q^m) u_m(P,Q).$$

Proof. Let α be as before; then

$$\alpha^m = \frac{v_m + u_m\sqrt{\Delta}}{2} = \frac{v_m + \sqrt{u_m^2\Delta}}{2} = \frac{v_m + \sqrt{v_m^2 - 4Q^m}}{2},$$

by Lemma 1, so

$$\alpha^m = \frac{P' + \sqrt{P'^2 - 4Q'}}{2},$$

where $P' = v_m(P,Q)$ and $Q' = Q^m$, and thus

$$(\alpha^m)^k = \frac{v_k(P',Q') + u_k(P',Q')\sqrt{P'^2 - 4Q'}}{2}$$
$$= \frac{v_k(P',Q') + u_k(P',Q')u_m(P,Q)\sqrt{\Delta}}{2}.$$

Now write

$$\alpha^{km} = \frac{v_{km}(P,Q) + u_{km}(P,Q)\sqrt{\Delta}}{2}$$

and compare the coefficients.

In the applications, Lucas sequences are often considered modulo a fixed modulus. If we choose a prime $p \neq 2$ for which the Legendre symbol $\left(\frac{\Delta}{p}\right) = -1$ then $\mathcal{O}_\Delta/p \cong \mathbf{F}_{p^2}$, the finite field of p^2 elements, via an isomorphism that we will denote by ϕ_p. The following lemma gives information about the order of α in $\mathcal{O}_\Delta/p$, and hence of $\phi_p(\alpha)$ in $\mathbf{F}_{p^2}$, which we will refer to in section 6.

Lemma 5. *Let* $\alpha = \frac{P \pm \sqrt{P^2-4Q}}{2}$ *and let* p *be an odd prime, with* $\left(\frac{P^2-4Q}{p}\right) = -1$. *Then:*

$$\alpha^{p+1} \equiv Q \bmod p.$$

Proof. In O_Δ/p:

$$\alpha^{p+1} = \alpha\left(\frac{v_1 + u_1\sqrt{\Delta}}{2}\right)^p = \alpha\left(\frac{v_1^p + u_1^p\sqrt{\Delta}^p}{2^p}\right) = \alpha\left(\frac{v_1 + \left(\frac{\Delta}{p}\right)u_1\sqrt{\Delta}}{2}\right) = \alpha\beta = Q$$

because

$$\Delta^{\frac{p-1}{2}} \equiv \left(\frac{\Delta}{p}\right) \equiv -1 \bmod p$$

by Euler's criterion.

3 LUC

In [21] the following cryptographic application of Lucas sequences was proposed, apparently independent of earlier publication in [13] and [14]. See also [25].

Public Key System (LUC). Each user publishes the product n of two large primes p and q, and an index e with $\gcd(e,(p^2-1)(q^2-1)) = 1$. The corresponding d such that $de \equiv 1 \bmod (p^2-1)(q^2-1)$ is kept secret (cf. [25: page 115]).

A message m is an integer satisfying $1 \leq m \leq n-1$ with $\gcd(m,n) = 1$. To encrypt a message m meant for some user, one looks up the user's n and e, and computes the encrypted message $y = v_e(m,1) \bmod n$ — i.e., P is equal to the message, and $Q = 1$. This computation can be carried out using the recurrence

given in Lemma 2 in $O(\log e)$ elementary operations on integers modulo n. To decrypt the message, the user calculates

$$v_d(y,1) \equiv v_d(v_e(m,1),1) \equiv v_{de}(m,1) \equiv m \bmod n$$

(cf. Lemma 4). The final identity holds because $\alpha^{de} \equiv \alpha$ modulo both p and q.

Alternatively, to use LUC as a signature scheme, the user's signature on a message m equals $v_d(m,1) \bmod n$, which can be verified by checking that $v_e(v_d(m,1),1) \equiv m \bmod n$.

Remarks. Our description of the choice of e and d is more general than the message-dependent choices from [21] or [25]; we refer to [21] and [25] for details. We would like to stress that the Lucas function using these message-dependent secret keys and the Lucas function using our choice of d are the same functions, since in both cases the inverse of $e \mapsto v_e(m,Q)$ is computed. In practical circumstances one would probably prefer to use message-dependent secret keys for efficiency reasons [24], for instance as follows.

Note that $v_{d(p)}(m,1) \equiv v_d(m,1) \bmod p$ if $d(p) \equiv d \bmod (p - (\frac{m^2-4}{p}))$ (use Lemma 5 if $(\frac{m^2-4}{p}) = -1$; otherwise use that $\alpha \in \mathbf{F}_p$). Signatures can therefore be generated substantially faster than computing $v_d(m,1) \bmod n$ by computing $v_{d(p)}(m,1) \bmod p$ and $v_{d(q)}(m,1) \bmod q$, followed by an application of the Chinese remainder theorem. However, no message-dependent d will be used in the sequel, because a message-independent d simplifies the analysis of LUC, and because in this paper we are not concerned with efficiency issues of LUC.

The choice $Q = 1$ is not essential for LUC as a public-key system: y could have been defined as $y = v_e(m,Q) \bmod n$, for some Q depending on the intended recipient, who can calculate $v_d(y,Q^e) \equiv v_d(v_e(m,Q),Q^e) \equiv v_{de}(m,Q) \equiv m$ modulo n. This would be slightly less efficient and offers no additional security. To use LUC as a signature system, however, either Q has to be equal to 1, or $Q^d \bmod n$ has to be included in the user's public key. Otherwise a verifier of the signature $v_d(m,Q)$ on message m would not be able to verify that $v_e(v_d(m,Q),Q^d)$ is indeed equivalent to m modulo n.

The signature $v_d(m,1)$ on message m can be used to generate signatures $v_d(v_k(m,1),1) \equiv v_{dk}(m,1) \equiv v_k(v_d(m,1),1) \bmod n$ on message $v_k(m,1)$ for any $k \geq 0$. This 'existential forgery' was mentioned in [2] and [6].

4 A chosen-message forgery for LUC

Let $n = pq$, e, and d be as above the public and secret data of some user, and let $Q = 1$. To forge the signature of this user on message m, an adversary could proceed as follows. First, integers a, b, c, s, and t are selected such that

$$bs - ct = 1, \qquad bs + ct = ae.$$

This can for instance be done by selecting c, h, and t such that $ct = (e-1)/2 + eh$ (note that e is odd), and selecting b and s such that $bs = 1 + ct$. It follows that $bs - ct = 1$, and that $bs + ct = 1 + 2ct = e + 2eh$, so that $a = 2h + 1$.

Next, the adversary calculates the messages $m_s = v_s(m,1) \bmod n$ and $m_t = v_t(m,1) \bmod n$ and obtains the user's signatures $v_d(m_s,1) \bmod n$ and $v_d(m_t,1) \bmod n$ on these messages. Finally, $v_d(m,1)$ is computed as

$$v_d(m,1) = v_b(v_d(m_s,1),1)v_c(v_d(m_t,1),1) - v_a(m,1) \bmod n.$$

The correctness follows from Lemma 4, the choice of a, b, c, s, t, m_s, and m_t, and from Lemma 2 with $k = dbs$, $\ell = dct$, and $Q = 1$:

$$\begin{aligned} v_b(v_d(m_s,1),1)v_c(v_d(m_t,1),1) &\equiv v_{dbs}(m,1)v_{dct}(m,1) \\ &\equiv v_{d(bs+ct)}(m,1) + v_{d(bs-ct)}(m,1) \\ &\equiv v_{dae}(m,1) + v_d(m,1) \\ &\equiv v_a(m,1) + v_d(m,1) \bmod n. \end{aligned}$$

Remarks. The mapping sending k to $v_k(m,1) \bmod n$ is not generally a random map into the message space (since it need not be surjective). As a consequence, the messages $v_s(m,1) \bmod n$ and $v_t(m,1) \bmod n$ that are to be signed are not always completely 'blind'.

If m, s, t and the signatures for $v_s(m,1) \bmod n$ and $v_t(m,1) \bmod n$ are given and if s, t, and e are pairwise relatively prime, then b, c satisfying $bs - ct = 1$ and $bs + ct \equiv 0 \bmod e$ can be found. Thus the signatures for m and $v_k(m,1)$ can be computed.

The choice of a, b, c, s, t is supposed to make it difficult for the user to find out which past signatures were used to make the forgery. The latter would be easy if we would have chosen $a = b = c = 1$, $s = (e+1)/2$, $t = (e-1)/2$, and

$$v_d(m,1) = v_d(m_s,1)v_d(m_t,1) - m \bmod n.$$

5 LUC and RSA

In the abstract and the introduction of [25] the authors of [25] announce a proof that LUC is cryptographically stronger than RSA. We have not been able to locate this proof in [25][5], and neither have we been able to derive such a proof ourselves. Here we offer some observations that might be pertinent to this matter.

Because $\alpha^{de} \equiv \alpha \bmod n$ and $u_1 = 1$, it follows from the second identity in Lemma 4 that

$$u_d(P,1) \equiv u_e(v_d(P,1),1)^{-1} \bmod n.$$

Thus $u_d(P,1)$ can be computed whenever $v_d(P,1)$ is known. Moreover, the following equation can be shown to hold by induction on k, using the recurrence relations for v_k and u_k:

[5] In [25: 3.4], however, the authors 'say, with confidence, that LUC is cryptographically stronger than RSA'.

$$2P^k \equiv v_k(P + P^{-1}, 1) + (P - P^{-1})u_k(P + P^{-1}, 1) \bmod n.$$

In particular, the above relations show that $P^d \bmod n$ can be derived once $v_d(P + P^{-1}, 1)$ is known.

To break an RSA-cryptogram $E(m)$, where $E(m) \equiv m^e \bmod n$ for some message m, it suffices to compute $E(m)^d \bmod n$, where e and d are as in the description of LUC, since $de \equiv 1 \bmod (p^2 - 1)(q^2 - 1)$ implies that $de \equiv 1 \bmod (p - 1)(q - 1)$. According to the above, this can be achieved if $v_d(E(m) + E(m)^{-1}, 1) \bmod n$ can be computed. Thus the RSA-cryptogram $E(m)$ can be broken if LUC can be broken for the message $E(m) + E(m)^{-1} \bmod n$.

This does not imply, however, that LUC is stronger than RSA. It is conceivable that LUC can only be broken for some particular set of messages, whereas RSA is secure. For instance, it might be the case that $v_d(X, 1)$ can only efficiently be derived from X, e, and n for X for which $\left(\frac{X^2-4}{p}\right) = \left(\frac{X^2-4}{q}\right) = -1$, where p and q are the prime factors of n. This would allow us to break 25% of all LUC-cryptograms, but since $\left(\frac{(X+X^{-1})^2-4}{p}\right) = \left(\frac{(X-X^{-1})^2}{p}\right) = 1$, the method cannot be used in the above manner to break RSA.

We are not aware of any further results in this direction.

6 LUCELG

In [26] the following cryptographic application of Lucas sequences was proposed.

Public Key System (LUCELG PK). A prime p and the start values P and $Q = 1$ are published, chosen such that $P^2 - 4Q \bmod p$ is a quadratic non-residue, and such that $v_\ell(P, Q) \not\equiv 2 \bmod p$ for any ℓ less than and dividing $p + 1$. Every user also chooses a private key x, and publishes the public key $y \equiv v_x(P, Q) \bmod p$ (cf. Lemma 2).

A message m is an integer satisfying $1 \leq m \leq p - 1$. To encrypt a message meant for some user, one looks up the user's y, chooses a secret k, which will also be an integer satisfying $1 \leq k \leq p - 1$, computes $G \equiv v_k(y, Q) \bmod p$, as well as $d_1 \equiv v_k(P, Q) \bmod p$ and $d_2 \equiv Gm \bmod p$. The encrypted message consists of the pair (d_1, d_2).

To decrypt the message, the user calculates

$$v_x(d_1, Q) \equiv v_x(v_k(P, Q), Q^k) \equiv v_{kx}(P, Q) \equiv G \bmod p,$$

inverts the result modulo p and recovers $m \equiv d_2 G^{-1} \bmod p$.

Remarks. Note that it seems essential in this scheme that $Q \equiv 1 \bmod p$: the recipient needs to know $Q^k \bmod p$ for the secret value k in order to be able to compute $v_{kx}(P, Q)$ from $v_k(P, Q)$ using the fourth lemma above. This can be achieved by taking $Q \equiv 1 \bmod p$; in [21; 22; 25; 26] it is assumed that $Q = 1$.

Let $\alpha = \frac{P+\sqrt{P^2-4Q}}{2}$; the condition that $v_\ell(P, Q) \not\equiv 2 \bmod p$ for proper divisors ℓ of $p + 1$ ensures that the multiplicative order of the image $\phi_p(\alpha) \in \mathbf{F}_{p^2}$

equals $p+1$. Namely, if $\phi_p(\alpha^n) = 1$ then $v_n(\alpha) \equiv 2 \bmod p$ and $u_n(\alpha) \equiv 0 \bmod p$, which does not happen for any proper divisor of $p+1$ by this condition. On the other hand, by Lemma 5 (with $Q \equiv 1 \bmod p$) the order divides $p+1$.

The condition that $(\frac{P^2-4Q}{p}) = -1$ (which is nowhere explicitly stated in [26]) guarantees that one is working in the the finite field $\mathbf{F}_{p^2}$ rather than $\mathbf{F}_p$; the latter contains a square root of P^2-4Q if the Legendre symbol equals 1 instead. In that case the attack described in the next section merely requires a discrete logarithm computation in $\mathbf{F}_p$. The recursive relations are still valid, but the order of α in O_Δ/p will be a divisor of $p-1$.

7 A subexponential time attack on LUCELG

Unfortunately, choosing $Q \equiv 1 \bmod p$ also provides the key to an attack on the proposed system: noting that

$$v_k(\alpha) = \alpha^k + \beta^k = \alpha^k + (\frac{Q}{\alpha})^k \equiv \alpha^k + \alpha^{-k} \bmod p$$

in this case, enables an adversary to obtain $\alpha^{\pm k}$ from v_k, since it is a root in $\mathbf{F}_{p^2}$ of the equation $z^2 - v_k z + 1 = 0$ (this is equivalent to deriving $\pm u_k(\alpha)$ and therefore $\alpha^{\pm k}$ from $v_k(\alpha)$ using Lemma 1(iii)). Then retrieving $\pm k$ from $\alpha^{\pm k}$ is a discrete logarithm problem in $\mathbf{F}_{p^2}$, which with the currently best available methods can be done in subexponential time $L_{p^2}[1/3; (64/9)^{1/3} + o(1)]$, for $p \to \infty$ (cf. [20]). Note that the sign of k does not matter, since $v_k = v_{-k}$ for all k when $Q = 1$, and that roots in $\mathbf{F}_{p^2}$ can be computed in expected polynomial time (cf. [3]). Other subexponential time methods to compute discrete logarithms in $\mathbf{F}_{p^2}$ can be found in [1; 5].

This implies that an adversary can derive x from y in subexponential time for any user, and decrypt all intercepted messages sent to that user. Alternatively, an adversary can decide only to derive k from the intercepted d_1, in subexponential time, after which G and thus m follow trivially from y and d_2.

Remarks. In [22; 26] an ElGamal-type signature scheme based on Lucas sequences was proposed (LUCELG DS). Since in this system both v_k and u_k are explicitly given, a direct analogue of the discrete logarithm attack on ElGamal (but here in $\mathbf{F}_{p^2}$) applies. Note that the 'double key size' problems of LUCELG DS as mentioned in [26] can be avoided if one uses Lemma 1(iii) to derive $\pm u_k$ from v_k. This would also avoid the serious weakness in LUCELG DS that is pointed out in [8]. Another variant of ElGamal based Lucas functions is discussed in [8]. The security of that system relies on the difficulty of computing discrete logarithms in $\mathbf{F}_p$.

In [22] a Diffie-Hellman-type key agreement scheme based on Lucas sequences was proposed (LUCDIF). Since LUCDIF again uses $Q = 1$, a subexponential attack similar to the one described above applies to it.

Acknowledgments. The authors are grateful to Eric Bach, Burt Kaliski, and Scott Vanstone, for their support of this article, which parallels similar remarks they sent or intended to send to the developers of Lucas-based cryptosystems. Christopher Skinner kindly communicated the effectiveness of our chosen-message attack in his 'message-dependent' implementation of LUC.

References

1. L. M. Adleman and J. DeMarrais, *A subexponential algorithm for discrete logarithms over all finite fields*, Proceedings Crypto'93, Lecture Notes in Comp. Sci. **773** (1994), 147–158.
2. E. Bach, *Comments on Peter Smith's LUC public-key encryption system*, manuscript, March 1993.
3. E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. **24** (1970), 713–735.
4. W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Info. Theory, vol IT-33 (1976), 644–654.
5. T. ElGamal, *A subexponential-time algorithm for computing discrete logarithms over $GF(p^2)$*, IEEE Trans. Info. Theory, vol IT-32 (1985), 469–472.
6. T. ElGamal and B. Kaliski, *Letter to the editor*, Dr. Dobb's Journal (May 1993), 10.
7. D. Gordon, *Discrete logarithms in $GF(p)$ using the number field sieve*, SIAM J. Disc. Math. **6** (1993), 124–138.
8. P. Horster, H. Petersen, and M. Michels, *Digital signature schemes based on Lucas functions*, University of Technology Chemnitz-Zwickau, Technical Report TR-95-1; to appear in: Communications and Multimedia Security, IT-Sicherheit '95, Joint working conference IFIP TC-6 TR-11 and Austrian Computer Society, Graz, Sept. 20–21, 1995.
9. C.-S. Laih, F.-K. Tu, and W.-C. Tai, *On the security of the Lucas function*, Information Processing Letters **53** (1995), 243–247.
10. A. K. Lenstra and H. W. Lenstra, Jr. (eds), *The development of the number field sieve*, Lecture Notes in Math. **1554**, Springer-Verlag, Berlin, 1993.
11. R. Lidl and W. B. Müller, *Permutation polynomials in RSA-cryptosystems*, Proceedings of Crypto'83, Plenum Press (1984), 293–301.
12. W. B. Müller, *Polynomial functions in modern cryptology*, Contributions to general Algebra 3, Proceedings of the Vienna conference (1985), 7–32.
13. W. B. Müller and W. Nöbauer, *Some remarks on public-key cryptosystems*, Studia Sci. Math. Hungar. **16** (1981), 71–76.
14. W. B. Müller and W. Nöbauer, *Cryptanalysis of the Dickson-scheme*, Proceedings of Eurocrypt'85, Springer (1985), 50–61.
15. NIST, *A proposed federal information processing standard for digital signature standard (DSS)*, Federal Register **56** (1991), 42980–42982.
16. W. Nöbauer, *Cryptanalysis of the Rédei-scheme*, Contributions to general Algebra **3**, Proceedings of the Vienna conference (1985), 255–264.
17. W. Nöbauer, *Cryptanalysis of a public-key cryptosystem based on Dickson-polynomials*, Mathematica Slovaca **38** (1989), 309–323.
18. H. Riesel, *Prime numbers and computer methods for factorization*, Progr. Math. **57**, Boston: Birkhauser, 1985.

19. R. L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), 120–126.
20. O. Schirokauer, *Using number fields to compute general discrete logarithms*, in preparation.
21. P. Smith, *LUC public-key encryption*, Dr. Dobb's Journal (January 1993), 44–49.
22. P. Smith, *Cryptography without exponentiation*, Dr. Dobb's Journal (April 1994), 26–30.
23. P. Smith, *Response to [6]*, Dr. Dobb's Journal (May 1993), 10–11.
24. P. Smith, Personal communication, February 1995.
25. P. J. Smith and M. J. J. Lennon, *LUC: a new public key system*, Proceedings of the Ninth IFIP Int. Symp. on Computer Security (1993), 103–117.
26. P. Smith and C. Skinner, *A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms*, Pre-proceedings Asiacrypt'94, 298–306.