Published in T. Matsumoto, ed., Proc. of the "4th ACM Conference on Computer and Communications Security", Zurich, 1-4 April 1997, pp. 135-141, ACM Press, 1997.

On the importance of securing your bins: The garbage-man-in-the-middle attack

Marc Joye

UCL Crypto Group, Dép. de Math.,
Université de Louvain, BELGIUM
E-mail: joye@agel.ucl.ac.be

Jean-Jacques Quisquater

UCL Crypto Group, Dép. d'Électricité,
Université de Louvain, BELGIUM
E-mail: jjq@dice.ucl.ac.be

URL: http://www.dice.ucl.ac.be/crypto/

Abstract

In this paper, we address the following problem: " Is it possible to weaken/attack a scheme when a (provably) secure cryptosystem is used? ". The answer is yes. We exploit weak error-handling methods. Our attack relies on the cryptanalyst being able to modify some ciphertext and then getting access to the decryption of this modified ciphertext. Moreover, it applies on many cryptosystems, including RSA, Rabin, LUC, KMOV, Demytko, ElGamal and its analogues, 3-pass system, knapsack scheme, etc...

1 Introduction

At Eurocrypt'96, Coppersmith, Franklin, Patarin and Reiter [4] presented a serious weakness in the basic protocol for encrypting related messages using the RSA public-key cryptosystem [24] with low exponent. Therefore, if an authority uses a RSA-based protocol to distribute secret keys among a group of users, a cryptanalyst can recover the secret keys of the users which have a relatively small public encryption key (typically less than 32 bits).

Suppose that Alice has a small public encryption key and that the authority wants to send her the secret key k_A, then the cryptanalyst does the following. He intercepts the ciphertext corresponding to k_A. So, since Alice does not receive the ciphertext, she asks the authority to re-send it. If the system is not well designed (for example, if the time or an ID number is included before the encryption), then the cryptanalyst applies the attack of Coppersmith *et al.* to recover the secret key from the two corresponding ciphertexts (the intercepted one and its retransmission).

Moreover, even if the authority is more malicious and chooses a clever protocol in order to prevent the previous attack, we will show that a cryptanalyst will be able to recover *all* the secrets keys, and not only the ones encrypted with a small exponent. Last but not least, our attack is of very general nature. It applies on many public-key cryptosystems, including RSA [24], Rabin [23, 30], LUC [27], KMOV [19], Demytko [8], ElGamal [12] and its analogues [28, 17], 3-pass system [10], knapsack scheme [25], etc...

The basic idea of our attack relies on the possibility to get access to the "bin" of the recipient. In fact, if the cryptanalyst intercepts, transforms and re-sends a ciphertext, then the corresponding plaintext will be meaningless when the authorized receiver (say, Alice) will decrypt it. So, Alice will discard it. If the cryptanalyst can get access to these discards, he will be able to recover the original plaintext if the transformation is done in a clever way. Such an attack was already be mounted against the RSA by Davida [5]. In many situations, we can get access to the discards, as for example,

- bad implementation of softwares or bad architectures;
- negligent secretaries;
- recovering of a previously deleted message, by a tool like the `<undelete>` command with MS-DOS;

another scenario is to ask the victim to sign the forged messages. Our working hypothesis are thus not unrealistic.

Consequently, the lesson of this paper is that the reader has to be very careful of using a given protocol to distribute secrets keys. In fact, the security of the used cryptosystem is not enough to guarantee the security of the transmitted data; we have also to *really* delete the discards, *i.e.* to secure our "bins".

The paper is organized as follows. In Section 2, we review the attack of Davida and give some (immediate) extensions. Next, we present our attack in Section 3. Section 4 deals with further results. Finally, we conclude in Section 5. The reader who is not familiar with Lucas sequences/elliptic curves over a ring and the resulting cryptosystems may consult the appendix.

2 Attack of Davida

Let p and q be two secret carefully chosen primes, and let $n = pq$ be the corresponding public RSA modulus. The pair (e, d) of public encryption/secret decryption keys of Bob are chosen according to $ed = 1 \pmod{\phi(n)}$.[†]

Suppose Alice wants to send a message m to Bob. Using the Bob's public encryption key, she computes $c = m^e \bmod n$, and sends it to Bob. Then, because only Bob knows the secret decryption key d, he can recover the message $m = c^d \bmod n$.

However, a cryptanalyst (Carol) can also recover the message as follows. She intercepts the ciphertext c, and

[†] ϕ denotes the Euler totient function, *i.e.* $\phi(n)$ is the number of positive integers $< n$ and relatively prime to n.

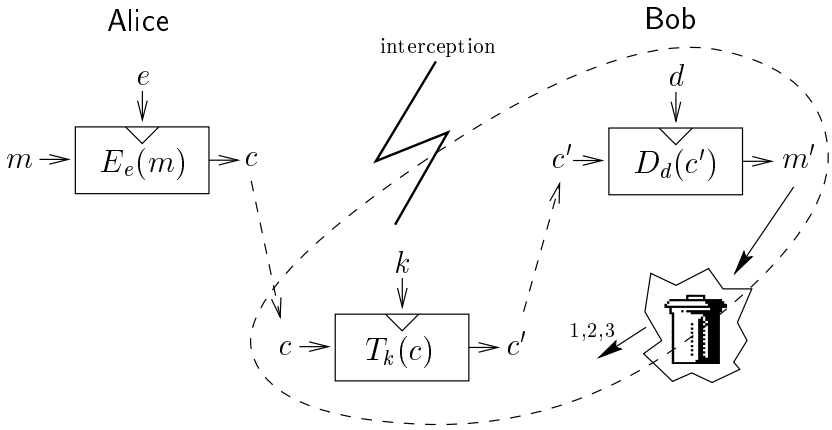

Figure 1: General description.

replaces it by $c' = ck^e \bmod n$ where k is a random number. Then, when Bob will decrypt c', he will compute $m' = c'^d \bmod n$. Since the message m' is meaningless, he will discard it. Consequently, if Carol can get access to m', she recovers the original message m by computing

$$m'k^{-1} = c'^d k^{-1} = c^d = m \pmod{n}.$$

This attack, first proposed by Davida [5], relies on the homomorphic nature of the RSA. Thus, it can easily be extended to other systems which have the same property [9]. So, DeMillo *et al.* [7] showed that the knapsack system is also vulnerable to this system. The same conclusion holds for the ElGamal cryptosystem and its variants based on Lucas sequences and elliptic curves over a finite field.

[Encryption] For sending a message m to Bob, Alice chooses a random r that is relatively prime to $p-1$. Then, she looks to the public key of Bob, $y = g^x \bmod p$ where x is the secret key of Bob. She computes the pair $(a = g^r \bmod p, b = my^r \bmod p)$ and sends it to Bob.

[Interception/modification] Carol intercepts the pair (a, b) and replaces it by $(a, b' = bk \bmod p)$ where k is a random. Next, she sends the modified pair to Bob.

[Decryption] Using his secret key x, Bob computes $m' = b'a^{-x} = mk \pmod{p}$. Since m' is meaningless, Bob discards it.

[Recovering] From m', Carol recovers the original message $m = m'k^{-1} \bmod p$.

Figure 2: Attacking ElGamal.

3 Our attack

3.1 General description

The basic attack of Davida does not apply to all cryptosystems. We need another tool:

Theorem 1 (Lagrange) *Let G be a (multiplicative) finite group of order n. Then each element a of G satisfies to $a^n = 1$.*

Let $E_e(\cdot)$, $D_d(\cdot)$ and $T_k(\cdot)$ be respectively the public encryption function, the corresponding secret decryption function and the cryptanalytic transformation function.

Now, imagine Alice wants to send the message m to Bob. She first computes the ciphertext $c = E_e(m)$, and sends it to Bob. Suppose that the cryptanalyst, Carol, intercepts the ciphertext c, and modifies it into $c' = T_k(c)$ according to the theorem of Lagrange. Next, Bob decrypts c', and gets $m' = D_d(c')$. Since the message m' has no meaning, Bob discards it. Finally, if Carol can get access to m', then she recovers the message m from c, k, c' and m' from a non-trivial relation.

We shall illustrate this attack on two cryptosystems that are not susceptible to the basic attack of Davida: LUC and KMOV/Demytko.* However, the same cryptanalysis remains valid for other systems.

3.2 Illustrations

3.2.1 Attacking LUC

For the LUC cryptosystem, the enciphering function and the deciphering function are respectively defined by

$$\begin{aligned} E_e &: L(D,n) \to L(D,n), \\ &(m,\cdot) \mapsto (c,\cdot) = (V_e(m,1),\cdot) \bmod n, \end{aligned} \tag{1}$$

$$\begin{aligned} D_d &: L(D,n) \to L(D,n), \\ &(c',\cdot) \mapsto (m',\cdot) = (V_d(c',1),\cdot) \bmod n. \end{aligned} \tag{2}$$

In order to modify the ciphertext c into c', the cryptanalyst uses the transformation function

$$\begin{aligned} T_k &: L(D,n) \to L(D,n), \\ &(c,\cdot) \mapsto (c',\cdot) = (V_k(c,1),\cdot) \bmod n. \end{aligned} \tag{3}$$

The non-trivial relation enabling the attack comes from the following proposition.

Proposition 1 *Let $\{V_k\}$ be the Lucas sequence with parameters P and $Q = 1$. Then*

$$V_k(P,1) = P^k - \sum_{\substack{i=1 \\ i \text{ odd}}}^{k-2} \binom{k}{\frac{k-i}{2}} V_i(P,1). \tag{4}$$

*See the appendix for a description of these cryptosystems.

So, it is possible to express recursively $V_k(P,1)$ as a polynomial of degree k in the indeterminate P.

Consequently, to recover the message m, the cryptanalyst, Carol, does the following.

[Lagrange's modification] Carol intercepts $c = V_e(m,1) \bmod n$ and replaces it by $c' = V_k(c,1) \bmod n$, where k is relatively prime to e.

[Recovery of m'] Next, she gets from Bob the value of m', the plaintext corresponding to c':

$$\begin{aligned} m' &= V_d(c',1) = V_{dk}(c,1) = V_{dke}(m,1) \\ &= V_k(m,1) \pmod n. \end{aligned}$$

[Non-trivial relation] She constructs the polynomials $\mathcal{P}, \mathcal{Q} \in \mathbb{Z}_n[x]$ given by

$$\mathcal{P}(x) = V_e(x,1) - c \text{ and } \mathcal{Q}(x) = V_k(x,1) - m',$$

for which m is a root. Then, she computes

$$\mathcal{R} = \gcd(\mathcal{P}, \mathcal{Q}),$$

that is with a very high probability a polynomial of degree 1. So, Carol obtains the value of m by solving $\mathcal{R}$ in x.

Figure 3: Attacking LUC.

3.2.2 Attacking KMOV/Demytko

To present the attack, we need to introduce the division polynomials (see [26], p. 105). They are recursively defined by

$$\begin{aligned} &\Psi_1 = 1, \quad \Psi_2 = 2y, \quad \Psi_3 = 3x^4 + 6ax^2 + 12bx - a^2, \\ &\Psi_4 = 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3), \\ &\Psi_{2m+1} = \Psi_{m+2}\Psi_m^3 - \Psi_{m-1}\Psi_{m+1}^2 \quad (m \geq 2), \\ &2y\,\Psi_{2m} = \Psi_m(\Psi_{m+2}\Psi_{m-1}^2 - \Psi_{m-2}\Psi_{m+1}^2) \quad (m \geq 2). \end{aligned}$$

Proposition 2 *Let $E_n(a,b)$ be an elliptic curve over the ring $\mathbb{Z}_n$. Define the polynomials Φ_k and ω_k by*

$$\begin{aligned} \Phi_k &= x\Psi_k^2 - \Psi_{k+1}\Psi_{k-1}, \\ 4y\,\omega_k &= \Psi_{k+2}\Psi_{k-1}^2 - \Psi_{k-2}\Psi_{k+1}^2. \end{aligned}$$

(a) Ψ_k, Φ_k, $y^{-1}\omega_k$ (for k odd) and $(2y)^{-1}\Psi_k$, Φ_k, ω_k (for k even) are polynomials in $\mathbb{Z}[a,b,x,y^2]$. Hence, by replacing y^2 by $x^3 + ax + b$, they will be considered as polynomials in $\mathbb{Z}[a,b,x]$.

(b) As polynomials in x,

$$\begin{aligned} \Phi_k(x) &= x^{k^2} + \textit{lower order terms}, \\ \Psi_k(x)^2 &= k^2 x^{k^2-1} + \textit{lower order terms} \end{aligned}$$

are relatively prime polynomials.

(c) If $P \in E_n(a,b)$, then

$$[k]P = \left(\frac{\Phi_k(P)}{\Psi_k(P)^2}, \frac{\omega_k(P)}{\Psi_k(P)^3} \right) \pmod n.$$

Corollary 1 *There is a univariate polynomial relation between the x-coordinates of a point P and any of its multiple kP over the elliptic curve $E_n(a,b)$ given by*

$$\Phi_k\Big(x(P)\Big) - x([k]P)\Psi_k\Big(x(P)\Big)^2 = 0 \pmod n.^{\ddagger} \qquad (5)$$

To illustrate the attack, we will only focus on the first coordinate of the points on $E_n(a,b)$. By (5), we have a polynomial relation between the first coordinate of a point and the first coordinate of any multiple of this point.

Let m_x be the message to be encrypted. This message will be represented by the point $M = (m_x, m_y)$ on the elliptic curve $E_n(a,b)$. Then, the cryptanalyst proceeds in a similar way as for LUC to recover the message m_x.

[Lagrange's modification] Carol intercepts $c_x = x([e]M)$ and replaces it by $c'_x = x([k]C)$, where k is relatively prime to e.

[Recovery of m'] Next, she gets from Bob the value of m'_x, the plaintext corresponding to c'_x:

$$m'_x = x([d]C') = x([dke]M) = x([k]M).$$

[Non-trivial relation] She constructs the polynomials $\mathcal{P}, \mathcal{Q} \in \mathbb{Z}_n[x]$ given by

$$\mathcal{P}(x) = \Phi_e(x) - c_x\Psi_e(x)^2$$

and

$$\mathcal{Q}(x) = \Phi_k(x) - m'_x\Psi_k(x)^2,$$

for which m_x is a root. Then, she computes

$$\mathcal{R} = \gcd(\mathcal{P}, \mathcal{Q}),$$

that is with a very high probability a polynomial of degree 1. So, Carol obtains the value of m_x by solving $\mathcal{R}$ in x.

Figure 4: Attacking KMOV/Demytko.

3.3 Analysis

In [22], Patarin estimates that the computation of a GCD may be done as long as the exponent of the polynomial is less than 32-bit long. So, unlike the basic attack of Davida against the RSA, from relation (4), our attack applies on LUC only if the public encryption exponent e has length less than 32 bits. Moreover, when Alice encrypts a message with the KMOV or Demytko cryptosystem with a public exponent e, the polynomial relation (5) is of order e^2 instead of e as in LUC. Therefore, our attack is useless against KMOV/Demytko if the encryption exponent e has length greater than 16 bits.

To overcome this drawback, the cryptanalyst (Carol) has to apply the attack in two times. We shall illustrate the technique on KMOV/Demytko. The notation is the same as used in paragraph 3.2.2.

Let $M = (m_x, m_y)$ and $C = [e]M = (c_x, c_y)$ be two points on the elliptic curve $E_n(a,b)$, where e is the public encryption key of Bob. Let m_x be the message to be encrypted. Then, the attack goes as follows. Carol intercepts c_x, and replaces it by $c'_{1,x} = x([k_1]C)$. Next, Bob computes

$$m'_{1,x} = x([d]C'_1) = x([dk_1e]M) = x([k_1]M). \qquad (6)$$

$^{\ddagger}x(P)$ denotes the x-coordinate of the point P.

Carol chooses k_2 (relatively prime to k_1), and sends $c'_{2,x} = x([k_2]C)$ to Bob. Then, Bob computes

$$m'_{2,x} = x([d]C'_2) = x([dk_2e]M) = x([k_2]M). \qquad (7)$$

Therefore, from relations (6) and (7), Carol forms the polynomials $\mathcal{P}$ and $\mathcal{Q} \in \mathbb{Z}_n[x]$ given by

$$\mathcal{P}(x) = \Phi_{k_1}(x) - m'_{1,x}\Psi_{k_1}(x)^2$$

and

$$\mathcal{Q}(x) = \Phi_{k_2}(x) - m'_{2,x}\Psi_{k_2}(x)^2,$$

for which m_x is a root. So, if k_1 and k_2 are "small" (typically less than 16-bit long), then by solving the polynomial $\mathcal{R} = \gcd(\mathcal{P}, \mathcal{Q})$, Carol obtains the message m_x.

4 Further results

4.1 Substituting the authority

Imagine we deal with a key distribution scheme. If the cryptanalyst intercepts the encrypted key c sent by the authority to Bob, and modifies it into c', then Bob will discover that the key is corrupted when he will decrypt it. Therefore, he will ask to the authority to re-send the key. If now, the cryptanalyst plays the role of the authority, *i.e.* if he sends the encrypted key c, then the authority will never know that a pirate knows the secret key of Bob. The cryptanalyst behaves thus transparently for the authority.

4.2 Concealing the cryptanalyst

The aim of the cryptanalyst is to modify the ciphertext c in such a way that Bob is not able to make the difference between his modification and noise (error of transmission) on the public channel. Consequently, he has to modify c in an apparently random way. So, he has to use "large" exponent for the transformation or to use the basic attack of Davida (when applicable).

We shall illustrate this topic on LUC. We use the same notation as in 3.2.1. Imagine that the cryptanalyst, Carol, chooses a small exponent k in order to speed up the computation of the GCD. Then, Bob can "prove" that somebody (*i.e.* Carol) modified the ciphertext c into c' by recovering k. He has just to compare (modulo n) $V_j(m,1)$ with m', for $j = 2, 3, \ldots, k$. To prevent this, Carol has to choose a relatively large exponent k. However, in order to recover the plaintext, she has to get access two or three times to the bin.

Let m be the message that Alice wants to send to Bob, and let $c = V_e(m,1) \bmod n$ be the corresponding ciphertext, where e is the public key of Bob. Then, the attack is the following.

Carol intercepts c, and replaces it by $c'_1 = V_{k_1}(c,1) \bmod n$. Bob receives c'_1, and decrypts it with its secret key d as $m'_1 = V_d(c'_1, 1) \bmod n$. Bob asks Alice to re-send c, and Carol sends $c'_2 = V_{k_2}(c,1) \bmod n$. When Bob computes $m'_2 = V_d(c'_2, 1) \bmod n$, he finds a meaningless message. So, he asks to Alice to send a third time the ciphertext c. Next, Carol sends $c'_3 = V_{k_3}(c,1) \bmod n$. Bob decrypts c'_3 to $m'_3 = V_d(c'_3, 1) \bmod n$, ...

Now, from the discards m'_i $(i = 1, 2(, 3))$, Carol can recover the original message m if k_1, k_2 and k_3 and correctly chosen. This can for instance be done by selecting

$$k_1 = rst, k_2 = ru \text{ and } k_3 = sv,$$

where r and s are small, and $\gcd(st, u) = \gcd(rt, v) = 1$. Note that t, u and v must be sufficiently large to disable Bob to distinguish noise with piracy on the public channel.

From our choices on the transformation keys k_i, Carol obtains

$$\begin{aligned} m'_1 = V_{rst}(m,1) &= V_{st}\Big(V_r(m,1),1\Big) \\ &= V_{rt}\Big(V_s(m,1),1\Big) \pmod n, \\ m'_2 = V_{ru}(m,1) &= V_u\Big(V_r(m,1),1\Big) \pmod n, \\ m'_3 = V_{sv}(m,1) &= V_v\Big(V_s(m,1),1\Big) \pmod n. \end{aligned}$$

Next, she forms the polynomials $\mathcal{Q}_{1_r}, \mathcal{Q}_2 \in \mathbb{Z}_n[y]$ given by

$$\mathcal{Q}_{1_r}(y) = V_{st}(y,1) - m'_1 \quad \text{and} \quad \mathcal{Q}_2(y) = V_u(y,1) - m'_2,$$

for which $V_r(m,1)$ is a root.
Hence, by computing $\mathcal{R} = \gcd(\mathcal{Q}_{1_r}, \mathcal{Q}_2)$, she gets (with a high probability) a polynomial of degree 1, for which $m_r = V_r(m,1) \bmod n$ is the root.

Now, if e is small, Carol recovers the original message m by constructing $\mathcal{P}, \mathcal{R} \in \mathbb{Z}_n[x]$ given by

$$\mathcal{P}(x) = V_e(x,1) - c \quad \text{and} \quad \mathcal{R}(x) = V_r(x,1) - m_r,$$

and by computing $\gcd(\mathcal{P}, \mathcal{Q})$, she recovers the original message m as explained before.

Otherwise, she constructs the polynomials $\mathcal{Q}_{1_s}, \mathcal{Q}_3 \in \mathbb{Z}_n[z]$ as

$$\mathcal{Q}_{1_s}(z) = V_{rt}(z,1) - m'_1 \quad \text{and} \quad \mathcal{Q}_3(z) = V_v(z,1) - m'_3,$$

for which $V_s(m,1)$ is a root.
Hence, by computing $\mathcal{S} = \gcd(\mathcal{Q}_{1_s}, \mathcal{Q}_3)$, she gets (with a high probability) a polynomial of degree 1, for which $m_s = V_s(m,1) \bmod n$ is the root. Next, from polynomials $\mathcal{R}, \mathcal{S} \in \mathbb{Z}_n[x]$ given by $\mathcal{R}(x) = V_r(x,1) - m_r$ and $\mathcal{S}(x) = V_s(x,1) - m_s$, she computes $\gcd(\mathcal{Q}, \mathcal{R})$ and recovers the message m.

Remark. It is possible to speed up the computation by using the ideas developed in [1] (see [16] for KMOV/Demytko). This will be done in a future work.

4.3 Combinations/Broadcast encryption

There are basically three ways for a cryptanalyst to recover a message

1. to force the retransmission;
2. to have a look in the bin;
3. to ask a signature.

Therefore, by combining these methods, it is possible to recover the message.

Furthermore, if the same message is broadcasted to several people, the security is compromised if only two or three persons are negligent (*i.e.* do not protect their bins).

4.4 Rabin-type cryptosystems

The Rabin-type cryptosystems [23, 30] have the advantage to be provably as intractable as factorization. However, since they are completely insecure against a chosen-ciphertext attack, our attack enables to recover the factorization of the public modulus n.

5 Conclusions

In this paper, we have shown that if one can get access to the "bin", then the security is compromised. We have illustrated our purpose with the Lagrange theorem. This is perhaps not the best tool. The basic aim of the paper was to warn the user that

> *"Even if a secure cryptosystem is used for a given application, this doesn't mean a pirate cannot mount a successful attack."*

This was shown to quite a lot of cryptosystems.

Acknowledgments

We are grateful to George Davida for sending a copy of his attack [5]. Thanks also to Moti Yung for suggesting the title.

References

[1] Bleichenbacher, D., Bosma, W., and Lenstra, A. K. Some remarks on Lucas-based cryptosystems. In *Advance in Cryptology – CRYPTO '95* (1995), D. Coppersmith, Ed., vol. 963 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 386–396.

[2] Bressoud, D. M. *Factorization and primality testing.* Undergraduate Texts in Mathematics. Springer-Verlag, 1989.

[3] Cohen, H. *A course in computational algebraic number theory*, vol. 138 of *Graduate Texts in Mathematics.* Springer-Verlag, 1993.

[4] Coppersmith, D., Franklin, M., Patarin, J., and Reiter, M. Low exponent RSA with related messages. In *Advance in Cryptology – EUROCRYPT '96* (1996), U. Maurer, Ed., vol. 1070 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–9.

[5] Davida, G. Chosen signature cryptanalysis of the RSA (MIT) public key cryptosystem. Tech. Rep. TR-CS-82-2, Dept. of Electrical Engineering and Computer Science, University of Wisconsin, Milwaukee, USA, Oct. 1982.

[6] DeLaurentis, J. M. A further weakness in the common modulus protocol for the RSA cryptoalgorithm. *Cryptologia 8*, 3 (July 1984), 253–259.

[7] DeMillo, R., Lynch, N. A., and Merritt, M. J. Cryptographic protocols. In *Proc. SIGACT Conf.* (1982).

[8] Demytko, N. A new elliptic curve based analogue of RSA. In *Advance in Cryptology – EUROCRYPT '93* (1993), T. Helleseth, Ed., vol. 765 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 40–49.

[9] Denning, D. E. Digital signatures with RSA and other public-key cryptosystems. *Communications of the ACM 27*, 4 (Apr. 1984), 388–392.

[10] Desmedt, Y., and Odlyzko, A. M. A chosen text attack on the RSA cryptosystem and some discrete logarithms schemes. In *Advance in Cryptology – CRYPTO '85* (1986), H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 516–521.

[11] Diffie, W., and Hellman, M. E. New directions in cryptography. *IEEE Transactions on Information Theory IT-26*, 6 (Nov. 1976), 644–654.

[12] ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory IT-31*, 4 (July 1985), 469–472.

[13] Håstad, J. On using RSA with low exponent in a public key network. In *Advance in Cryptology – CRYPTO '85* (1986), H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 404–408.

[14] Husemöller, D. *Elliptic curves*, vol. 111 of *Graduate Texts in Mathematics.* Springer-Verlag, 1987.

[15] Joye, M., and Quisquater, J.-J. Protocol failures for RSA-like functions using Lucas sequences and elliptic curves over a ring. Pre-proceedings of the 1996 Cambridge Workshop on Security Protocols, Apr. 1996.

[16] Kaliski Jr, B. S. A chosen message attack on Demytko's elliptic curve cryptosystem. To appear in Journal of Cryptology.

[17] Koblitz, N. Elliptic curve cryptosystems. *Mathematics of Computation 48* (1987), 203–209.

[18] Koblitz, N. *A course in number theory and cryptography*, 2nd ed., vol. 114 of *Graduate Texts in Mathematics.* Springer-Verlag, 1994.

[19] Koyama, K., Maurer, U. M., Okamoto, T., and Vanstone, S. A. New public-key schemes based on elliptic curves over the ring $\mathbb{Z}_n$. In *Advance in Cryptology – CRYPTO '91* (1991), J. Feigenbaum, Ed., vol. 576 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 252–266.

[20] Menezes, A. J. *Elliptic curve public key cryptosystems.* Kluwer Academic Publishers, 1993.

[21] Miller, V. S. Use of elliptic curves in cryptography. In *Advance in Cryptology – CRYPTO '85* (1986), H. C. Williams, Ed., vol. 218 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 417–426.

[22] Patarin, J. Some serious protocol failures for RSA with exponent e of less than $\simeq 32$ bits. Presented at the conference of cryptography, CIRM Luminy, France, Sept. 1995.

[23] Rabin, M. O. Digital signatures and public-key functions as intractable as factorization. Tech. Rep. MIT/LCS/TR-212, MIT Laboratory for Computer Science, Jan. 1979.

[24] Rivest, R. L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*, 2 (Feb. 1978), 120–126.

[25] Shamir, A. A fast signature scheme. Tech. Rep. MIT/LCS/TM-107, MIT Lab. for Computer Science, Cambridge, Mass., July 1978.

[26] Silverman, J. *The arithmetic of elliptic curves*, vol. 106 of *Graduate Texts in Mathematics.* Springer-Verlag, 1986.

[27] SMITH, P. LUC public-key encryption. *Dr. Dobb's Journal* (Jan. 1993), 44–49.

[28] SMITH, P., and SKINNER, C. A public-key cryptosystem and a digital signature based on the Lucas function analogue to discrete logarithms. In *Advance in Cryptology – ASIACRYPT '94* (1995), J. Pieprzyk, Ed., vol. 917 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 357–364.

[29] SMITH, P. J., and LENNON, M. J. J. LUC: A new public key system. In *Ninth IFIP Symposium on Computer Security* (1993), E. G. Douglas, Ed., Elsevier Science Publishers, pp. 103–117.

[30] WILLIAMS, H. C. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory IT-26*, 6 (Nov. 1980), 726–729.

A Lucas sequences

In 1993, Smith [27] proposed to use the Lucas sequences in order to construct a public-key cryptosystem.

The reader who is not familiar with Lucas sequences may found an elementary introduction in [2].

A.1 Definition

Let D be an integer congruent to 0 or 1 modulo 4, which is a non-square, and let P be an integer with the same parity as D such that 4 divides $P^2 - D$. The *Lucas sequences* $\{U_i\}_{i\geq 0}$ *and* $\{V_i\}_{i\geq 0}$ are defined by

$$V_i + U_i\sqrt{D} = 2^{1-i}(P + \sqrt{D})^i. \tag{8}$$

It can easily be shown that the numbers U_i and V_i satisfy the following relations:

$$\begin{aligned} U_0 = 0, U_1 = 1, V_0 = 2, V_1 = P, \\ U_{i+j} = U_i V_j - Q^j U_{i-j}, \\ V_{i+j} = V_i V_j - Q^j V_{i-j}, \end{aligned}$$

where $Q = (P^2 - D)/4$.

Moreover, if we take $Q = 1$ in the definition of the Lucas sequences and if n is relatively prime to $2D$, then we call $L(D, n)$ the group of Lucas sequences (see [2], p. 196). The elements of $L(D, n)$ are the pairs (V_i, U_i) modulo n obtained from the Lucas sequences $\{V_i\}$ and $\{U_i\}$ with parameter $Q = 1$. The identity element is $(2, 0)$. Two elements $(V_k, U_k), (V_m, U_m) \in L(D, n)$ are composed according to the law ∂ defined by $(V_k, U_k)\partial(V_m, U_m) = (V_{k+m}, U_{k+m})$.

Theorem 2 *If $p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$ is the prime factorization of n, then the order of $L(D, n)$ is given by*

$$\#L(D, n) = (p_1 - \left(\frac{D}{p_1}\right))p_1^{e_1 - 1}(p_2 - \left(\frac{D}{p_2}\right))p_2^{e_2 - 1} \cdots (p_r - \left(\frac{D}{p_r}\right))p_r^{e_r - 1}.$$ [§]

Corollary 2 *Let $\Psi(p) = p - (D/p)$, where p is a prime which does not divide $2D$. Then, for any integer k,*

$$V_{k\Psi(p)+1}(P, 1) \equiv P \pmod{p}.$$

A.2 LUC

The LUC cryptosystem is based on the following proposition.

Proposition 3 *Let $V_i(P, 1)$ be the i^{th} terms of the Lucas sequence $\{V_i\}$ with parameters P and $Q = 1$. Then,*

$$V_{mk}(P, 1) = V_m\Big(V_k(P, 1), 1\Big).$$

Each user chooses two secret large primes p and q, and publishes their product $n = pq$. Next, he chooses a public encryption key e which is relatively prime to $(p-1)$, $(p+1)$, $(q-1)$ and $(q+1)$.

To send an encrypted message m to Bob, Alice looks to Bob's public key e, and using the Lucas sequence $\{V_i\}$,

[§] (D/p) denotes the Legendre symbol and is equal to 1 or -1 if D is a quadratic residue modulo p or not.

she forms the ciphertext $c = V_e(m,1) \bmod n$. Since Bob knows the factors p and q, he is able to compute the secret decryption key d according to $ed = 1 \bmod \Psi(n)$, where $\Psi(n) = \mathrm{lcm}(p-(D/p), q-(D/q))$. Therefore, he recovers the message m by computing

$$V_d\Big(V_e(m,1),1\Big) = V_{ed}(m,1) = V_1(m,1) = m \pmod n.$$

B Elliptic curves

In 1991, Koyama, Maurer, Okamoto and Vanstone [19] proposed new trapdoor one-way functions (TOFs) based on elliptic curves over the ring $\mathbb{Z}_n$. We call KMOV the resulting cryptosystem. Two years later, Demytko [8] proposed another cryptosystem based on the same structure.

For an introduction to elliptic curves, the reader may consult [20, 18]. A deeper study may be found in [26].

B.1 Elliptic curves over $\mathbb{Z}_n$

Let $\mathbb{F}_p$ be a prime field of characteristic $p \neq 2, 3$, and let $a, b \in \mathbb{F}_p$ such that $4a^2+27b^3 \neq 0$. An elliptic curve over $\mathbb{F}_p$ with parameters a and b is the set of points (x,y) satisfying the Weierstraß equation

$$y^2 = x^3 + ax + b, \tag{9}$$

together with a special point $\mathcal{O}$ called the point at infinity. Such a curve will be denoted by $E_p(a,b)$.

Let $P, Q \in E_p(a,b)$, ℓ be the line connecting P and Q (tangent line if $P = Q$), and T be the third point of intersection of ℓ with $E_p(a,b)$. Let ℓ' be the line connecting T and $\mathcal{O}$. Then $P+Q$ is the point such that ℓ' intersects $E_p(a,b)$ at $T, \mathcal{O}$ and $P+Q$. This composition law makes $E_p(a,b)$ into an Abelian group with identity element $\mathcal{O}$.

An elliptic curve over the ring $\mathbb{Z}_n$ is defined in the same way, except that all computations are done modulo n instead of modulo p. However the resulting structure is not a group, but a proposition similar to the Lagrange theorem holds.

Proposition 4 *Let $n = pq$, where p and q are prime. Let $E_n(a,b)$ be an elliptic curve such that $\gcd(4a^3+27b^2, n) = 1$, and let*

$$N_n = \mathrm{lcm}(\#E_p(a,b), \#E_q(a,b)). \tag{10}$$

Then, for any $P \in E_n(a,b)$, and any integer k,

$$[kN_n + 1]P = P$$

on $E_n(a,b)$.[¶]

Remark. As mentioned in [19], it is also possible to define an elliptic curve over a ring such that the resulting structure is a group.

B.2 KMOV/Demytko

The KMOV and the Demytko cryptosystems are both based on elliptic curves over the ring $\mathbb{Z}_n$, where n is the product of two primes p and q.

Imagine Alice wants to send a message m to Bob. She first represents the message m by a publicly known way as a point $M = (m_x, m_y)$ of the elliptic curve $E_n(a,b)$. Then, by using the public encryption key of Bob, she computes $C = (c_x, c_y) = [e]M$ over $E_n(a,b)$.

To recover the message m, Bob computes $M = [d]C$ over $E_n(a,b)$ with his secret decryption d according to $ed = 1 \pmod{N_n}$.

In the KMOV cryptosystem, the primes p and q are both congruent to 2 modulo 3, and the parameter a is equal to 0. In that case, we can show that

$$N_n = \mathrm{lcm}(\#E_p(0,b), \#E_q(0,b)) = \mathrm{lcm}(p+1, q+1).$$

Since N_n does not depend on b, the parameter b is chosen according to

$$b = m_y^2 - m_x^3 \bmod n.$$

With the Demytko cryptosystem, in order to decrypt $C = (c_x, c_y)$, Bob computes $[d_i]C$, where the decryption key $d = d_i$ is chosen so that

$$ed_i \equiv 1 \pmod{N_{n,i}} \quad (i = 1, \ldots, 4),$$

$$\text{with } \begin{cases} N_{n,1} = \mathrm{lcm}(\#E_p(a,b), \#E_q(a,b)) \\ \qquad \text{if } (w/p) = 1 \text{ and } (w/q) = 1 \\ N_{n,2} = \mathrm{lcm}(\#E_p(a,b), \#\overline{E_q(a,b)}) \\ \qquad \text{if } (w/p) = 1 \text{ and } (w/q) \neq 1 \\ N_{n,3} = \mathrm{lcm}(\#\overline{E_p(a,b)}, \#E_q(a,b)) \\ \qquad \text{if } (w/p) \neq 1 \text{ and } (w/q) = 1 \\ N_{n,4} = \mathrm{lcm}(\#\overline{E_p(a,b)}, \#\overline{E_q(a,b)}) \\ \qquad \text{if } (w/p) \neq 1 \text{ and } (w/q) \neq 1 \end{cases},$$

and $w = c_x^3 + ac_x + b \bmod n$.[††]

Remarks. 1) In the KMOV cryptosystem, it is also possible to work on the elliptic curve $E_n(a,0)$, if one chooses p and q both congruent to 3 modulo 4.
2) The computation of the second coordinate can be avoided if the algorithm described in [2, pp. 211-216] is used.

[¶] $[k]P$ means $\underbrace{P+P+\cdots+P}_{k \text{ times}}$.

[††] $\overline{E_p(a,b)}$ denotes the complementary group of $E_p(a,b)$, *i.e.* the set of points satisfying Weierstraß equation (9) together with $\mathcal{O}$, where y is of the form $u\sqrt{v}$ with v is a fixed quadratic non-residue modulo p and $u \in \mathbb{F}_p$.