Published in M. Darnell, Ed., *Cryptography and Coding*, vol. 1355 of *Lecture Notes in Computer Science*, pp. 155–160, Springer-Verlag, 1997.

RSA-type Signatures in the Presence of Transient Faults

Marc Joye[1], Jean-Jacques Quisquater[2], Feng Bao[3] and
Robert H. Deng[3]

[1] UCL Crypto Group, Dept of Mathematics, University of Louvain
Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium
Email: `joye@agel.ucl.ac.be`

[2] UCL Crypto Group, Microelectronics Labs, University of Louvain
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
Email: `jjq@dice.ucl.ac.be`

[3] Institute of Systems Science, National University of Singapore
Kent Ridge, Singapore 119597
Email: {`baofeng, deng`}`@iss.nus.sg`

Abstract. In this paper, we show that the presence of transient faults can leak some secret information. We prove that only one faulty RSA-signature is needed to recover one bit of the secret key. Thereafter, we extend this result to Lucas-based and elliptic curve systems.

Keywords. RSA, Lucas sequences, elliptic curves, transient faults.

1 Introduction

At the last Workshop on Security Protocols, Bao, Deng, Han, Jeng, Narasimhalu and Ngair from the Institute of Systems Science (Singapore) exhibited new attacks against several cryptosystems [2]. These attacks exploit the presence of transient faults. By exposing a device to external constraints, one can induce some faults with a non-negligible probability [1].

In this paper, we show that these attacks are of very general nature and remain valid for cryptosystems based on other algebraic structures. We will illustrate this topic on the Lucas-based and elliptic curve cryptosystems. Moreover, we will focus on the signatures generation, reducing the number of required signatures for a successful attack to one.

The paper is organized as follows. In Section 2, we recall the RSA-type signature schemes. Next, we present the basic attack in Section 3, and extend it to Lucas sequences and elliptic curves. In Section 4, we give some further results. Finally, we conclude in Section 5.

2 RSA-type signatures

Let $n = pq$ be the product of two large primes. Let G be the (multiplicatively written) group given by the direct product of groups

$$G = G_p \times G_q. \tag{1}$$

The public verification key v is chosen relatively prime to $\#G$, and the secret signature key s is chosen according to

$$vs \equiv 1 \pmod{\#G}. \tag{2}$$

To sign a message $M \in G$, Bob computes the corresponding signature

$$S = M^s \tag{3}$$

with his secret key s. Then, he transmits the pair (M, S) to Alice. In order to verify the signature, Alice checks if $S^v = M$, where v is the public key of Bob.

Remark 1. For security and efficiency purposes, many variations of this signature scheme were proposed ; however, since our attack does not depend on these modifications, we decided to work with this elementary model.

The original RSA cryptosystem [9] works in the group $G = \mathbb{Z}_n^* \,(= \mathbb{F}_p^* \times \mathbb{F}_q^*)$, the group of units of integers modulo n. It was later extended to Lucas sequences by Smith and Lennon [11] to produce LUC. In this case, we have $G = \{V_j(P, 1) = r^j + r^{-j} \bmod n\}$ where r is a root of $x^2 - Px + 1 = 0$. Elliptic curves were also proposed to implement analogues of RSA. The first scheme, called KMOV, was presented by Koyama, Maurer, Okamoto and Vanstone [7]. It is based on elliptic curves of the form $E_n(0, b)\,(= E_p(0, b) \times E_q(0, b))$ or $E_n(a, 0)\,(= E_p(a, 0) \times E_q(a, 0))$ in order to have a message-independent group order. Another scheme was proposed by Demytko [5]. It has the particularity to only use the first coordinate of the points of elliptic curves. It relies on the fact that an integer x must be the x-coordinate of a fixed elliptic curve $E_p(a, b)$ or of its twist $\overline{E_p(a, b)}$.

We refer to the original papers for a complete description of these systems.

3 Transient faults

We assume that an error occurs during the computation of the signature S. More precisely, if $s = \sum_{i=0}^{t-1} s_i \, 2^i$ denotes the binary expansion of the secret key s, we suppose that the j^{th} bit of s, namely s_j, flips to its complementary value. Therefore, Bob will obtain the faulty signature

$$\hat{S} = M^{\hat{s}} \tag{4}$$

instead of S, where

$$\hat{s} = \begin{cases} s + 2^j & \text{if } s_j = 0, \\ s - 2^j & \text{if } s_j = 1. \end{cases} \tag{5}$$

3.1 Attacking the RSA

Let $S = M^s \bmod n$ and $\hat{S} = M^{\hat{s}} \bmod n$ be the correct and the faulty signatures corresponding to message M, respectively. Mimicking the attack in [2], we find the flipped bit of s by computing

$$\frac{\hat{S}}{S} \equiv M^{\hat{s}-s} \equiv \begin{cases} M^{2^j} \pmod{n} & \text{if } s_j = 0, \\ \frac{1}{M^{2^j}} \pmod{n} & \text{if } s_j = 1. \end{cases} \tag{6}$$

However, this formulation is not optimal in signature context. Putting Eq. (6) to the v, we obtain

$$\frac{\hat{S}^v}{M} \equiv \begin{cases} (M^v)^{2^j} \pmod{n} & \text{if } s_j = 0, \\ \frac{1}{(M^v)^{2^j}} \pmod{n} & \text{if } s_j = 1. \end{cases} \tag{7}$$

So only the faulty signature $\hat{S}$ is required to recover the flipped bit. The attack can thus be summarized as follows. The attacker randomly chooses a message M. He computes $F := M^v \bmod n$ and $\alpha_j := F^{2^j} \bmod n$. Then, inducing a physical effort, he asks the device to sign message M. If one bit of s has flipped, he easily recover it by comparing $\hat{S}^v/M$ to α_j and $1/\alpha_j$.

3.2 Attacking LUC

The attack on LUC works similarly. The attacker chooses a random message M. He computes $F := V_v(M, 1) \bmod n$, $G := (M^2 - 4)U_v(M, 1) \bmod n$, $\alpha_j := V_{2^j}(F, 1) \bmod n$ and $\beta_j := U_{2^j}(F, 1) \bmod n$. Now, inducing a external effort to the signing device, he gets the faulty signature of message M,

$$\hat{S} = V_{\hat{s}}(M, 1) \bmod n.$$

Finally, he finds the flipped bit of s as

$$2V_v(\hat{S}, 1) \equiv \begin{cases} \alpha_j M + \beta_j G \pmod{n} & \text{if } s_j = 0, \\ \alpha_j M - \beta_j G \pmod{n} & \text{if } s_j = 1. \end{cases} \tag{8}$$

Proof. From the properties of Lucas sequences [3, Chapter 12], it follows that

$$\begin{aligned} 2V_v(\hat{S}, 1) &\equiv 2V_v\big(V_{\hat{s}}(M, 1), 1\big) \equiv V_{v(\hat{s}-s+s)}(M, 1) \equiv 2V_{v(\hat{s}-s)+1}(M, 1) \\ &\equiv V_{v(\hat{s}-s)}(M, 1)V_1(M, 1) + \Delta U_{v(\hat{s}-s)}(M, 1)U_1(M, 1) \\ &\equiv V_{\hat{s}-s}\big(V_v(M, 1), 1\big)M + (M^2 - 4)U_v(M, 1)U_{\hat{s}-s}\big(V_v(M, 1), 1\big) \\ &\equiv V_{\hat{s}-s}(F, 1)M + U_{\hat{s}-s}(F, 1)G \pmod{n}. \end{aligned}$$

Furthermore, since $V_{-k}(P, 1) = V_k(P, 1)$ and $U_{-k}(P, 1) = -U_k(P, 1)$, we have the required result. □

An efficient implementation of Lucas sequences may be found in [6]. Notice also that the computation of α_j and β_j is not expensive. They can recursively be evaluated by

$$\begin{cases} \alpha_0 = F \\ \alpha_j = \alpha_{j-1}^2 - 2 \bmod n \end{cases} \quad \text{and} \quad \begin{cases} \beta_0 = 1 \\ \beta_j = \alpha_{j-1}\beta_{j-1} \bmod n \end{cases}. \tag{9}$$

3.3 Attacking KMOV

In the KMOV system, the messages are represented as points of elliptic curves. Let $\mathbf{M} = (M_1, M_2)$ be the message being signed, and let $\mathbf{S} = [s]\mathbf{M}$ and $\hat{\mathbf{S}} = [\hat{s}]\mathbf{M}$ respectively be the correct and the faulty signatures. We still assume that bit j of s flipped during the computation of the signature, *i.e.* $\hat{s} = s \pm 2^j$.

To recover this flipped bit, the attacker has just to compare $[v]\hat{\mathbf{S}} - \mathbf{M}$ to $[\hat{s} - s]\,([v]\mathbf{M})$. More precisely,

$$[v]\hat{\mathbf{S}} - \mathbf{M} = \begin{cases} [2^j](F_1, F_2) & \text{if } s_j = 0 \\ [2^j](F_1, -F_2) & \text{if } s_j = 1 \end{cases}, \tag{10}$$

where $(F_1, F_2) := [v]\mathbf{M}$.

Proof. Straightforwardly, we have

$$\begin{aligned} [v]\hat{\mathbf{S}} &= [v\hat{s}]\mathbf{M} = [v(\hat{s} - s + s)]\mathbf{M} = [v(\hat{s} - s)]\mathbf{M} + \mathbf{M} \\ &= [\hat{s} - s]\,([v]\mathbf{M}) + \mathbf{M}. \end{aligned}$$

Moreover, if $\mathbf{P} = (P_1, P_2)$ is a point on an elliptic curve, then its inverse is given by $-\mathbf{P} = (P_1, -P_2)$. □

3.4 Attacking Demytko

The Demytko's system only uses the x-coordinate of points of elliptic curves. Let M the message to be signed. This message is represented as a point $\mathbf{M} = (M, \cdot)$, or equivalently $M = x(\mathbf{M})$. As before, we suppose that a bit of s flipped during the computation of the signature. So, the resulting signature is $\hat{S}$, which is represented by $\hat{\mathbf{S}} = (\hat{S}, \cdot)$.

It is useful to introduce some notation. Let $\mathbf{P} = (x, y)$ be a point of an elliptic curve and let $\mathbf{Q} = [k]\mathbf{P}$. Using division polynomials [8, Chapter II], we can compute $x(\mathbf{Q}) = x([k]\mathbf{P})$ and $y(\mathbf{Q})/y(\mathbf{P}) = y([k]\mathbf{P})/y(\mathbf{P})$ from the only knowledge of $x = x(\mathbf{P})$ [8, Theorem 2.1]. These latter functions will respectively be denoted $\mathcal{X}_k(x)$ and $\mathcal{Y}_k(x)$.

To recover the flipped bit of s, the attacker computes $F := x([v]\mathbf{M}) = \mathcal{X}_v(M)$, $G := y([v]\mathbf{M})/y(\mathbf{M}) = \mathcal{Y}_v(M)$, $H := M^3 + aM + b \bmod n$, $\alpha_j := \mathcal{X}_{2^j}(F)$ and $\beta_j := \mathcal{Y}_{2^j}(F)$. Then,

$$\mathcal{X}_v(\hat{S}) + M \equiv \begin{cases} \frac{(\beta_j G - 1)^2 H}{(\alpha_j - M)^2} - \alpha_j \pmod{n} & \text{if } s_j = 0, \\ \frac{(\beta_j G + 1)^2 H}{(\alpha_j - M)^2} - \alpha_j \pmod{n} & \text{if } s_j = 1. \end{cases} \tag{11}$$

Proof. The chord-and-tangent addition formulae on elliptic curves [10, Algorithm 2.3] yields

$$\begin{aligned} \mathcal{X}_v(\hat{S}) &\equiv x([v]\hat{\mathbf{S}}) \equiv x\left([\hat{s} - s]([v]\mathbf{M}) + \mathbf{M}\right) \\ &\equiv \left[\frac{y([v(\hat{s} - s)]\mathbf{M}) - y(\mathbf{M})}{x([v(\hat{s} - s)]\mathbf{M}) - x(\mathbf{M})}\right]^2 - x([v(\hat{s} - s)]\mathbf{M}) - x(\mathbf{M}) \end{aligned}$$

$$\begin{aligned}&\equiv \frac{M^3+aM+b}{\left(x\left([\hat{s}-s]([v]\mathbf{M})\right)-M\right)^2}\left[\frac{y\left([\hat{s}-s]([v](\mathbf{M})\right)}{y(\mathbf{M})}-1\right]^2\\ &\qquad -x\left([\hat{s}-s]([v]\mathbf{M})\right)-M\\ &\equiv \frac{H\left[\frac{y([\hat{s}-s]([v](\mathbf{M}))}{y([v]\mathbf{M})}\,\frac{y([v]\mathbf{M})}{y(\mathbf{M})}-1\right]^2}{(\mathcal{X}_{\hat{s}-s}(F)-M)^2}-\mathcal{X}_{\hat{s}-s}(F)-M\\ &\equiv \frac{H\big(\mathcal{Y}_{\hat{s}-s}(F)G-1\big)^2}{(\mathcal{X}_{\hat{s}-s}(F)-M)^2}-\mathcal{X}_{\hat{s}-s}(F)-M \pmod{n}.\end{aligned}$$

Hence, since $\mathcal{X}_{-k}(x)=\mathcal{X}_k(x)$ and $\mathcal{Y}_{-k}(x)=-\mathcal{Y}_k(x)$, the proof is complete. □

4 Further results

4.1 Generalizing the number of faulty bits

Assume that two bits of s are faulty when performing the signature of a message M with the RSA. Then, the signature is $\hat{S}=M^{\hat{s}} \bmod n$, where

$$\hat{s}=s\pm 2^j \pm 2^k. \tag{12}$$

As before, the attacker computes $F:=M^v \bmod n$ and $\alpha_j:=F^{2^j} \bmod n$, and compares $\hat{S}^v/M$ to $(\alpha_j)^{\pm 1}(\alpha_k)^{\pm 1}$ until a match is found. In this case, he finds two bits of the secret key s.

This method naturally extends to multiple faulty bits and to Lucas-based and elliptic curve systems.

4.2 Davida's attack

Our attack also applies for encryption process. This can been considered as a special case of the Davida's attack [4]. We shall illustrate this attack on RSA, but it remains valid for Lucas-based and elliptic curve systems.

Let e and d be a pair of public encryption key and secret decryption key related by

$$ed\equiv 1 \pmod{\operatorname{lcm}(p-1,q-1)}. \tag{13}$$

The attacker chooses a random message M, computes the ciphertext $C=M^e \bmod n$ with the public key e of Bob. Then, inducing a external constraint, he asks Bob to decipher C. If Bob does so, he obtains $\hat{M}=C^{\hat{d}} \bmod n$. Since $\hat{M}$ is meaningless, Bob will discard it. Suppose that the attacker can get access to this discard and that only one bit of d is faulty, i.e. $\hat{d}=d\pm 2^j$. So,

$$\frac{\hat{M}^e}{C}\equiv\begin{cases}(C^e)^{2^j} \pmod{n} & \text{if } d_j=0,\\ \frac{1}{(C^e)^{2^j}} \pmod{n} & \text{if } d_j=1.\end{cases} \tag{14}$$

5 Conclusion

We have shown that the presence of transient faults gives the bit-values of the secret signature key s in RSA-type signature schemes. As in [2], this attack may of course be generalized to discrete log based cryptosystems.

References

1. Anderson, R., and Kuhn, M. Tamper resistance – a cautionary note. In *Proceedings of the Second USENIX Workshop on Electronic Commerce* (1996), USENIX Association, pp. 1–11.
2. Bao, F., Deng, R. H., Han, Y., Jeng, A., Narasimhalu, A. D., and Ngair, T. Breaking public key cryptosystems on tamper resistant devices in the presence of faults. In *Pre-proceedings of the 1997 Security Protocols Workshop* (1997).
3. Bressoud, D. M. *Factorization and primality testing.* Undergraduate Texts in Mathematics. Springer-Verlag, 1989.
4. Davida, G. Chosen signature cryptanalysis of the RSA (MIT) public key cryptosystem. Tech. Report TR-CS-82-2, Dept. of Electrical Engineering and Computer Science, University of Wisconsin, Milwaukee, USA, Oct. 1982.
5. Demytko, N. A new elliptic curve based analogue of RSA. In *Advances in Cryptology – EUROCRYPT '93* (1994), T. Helleseth, Ed., vol. 765 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 40–49.
6. Joye, M., and Quisquater, J.-J. Efficient computation of full Lucas sequences. *Electronics Letters 32*, 6 (Mar. 1996), 537–538.
7. 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 *Advances in Cryptology – CRYPTO '91* (1992), J. Feigenbaum, Ed., vol. 576 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 252–266.
8. Lang, S. *Elliptic curves: Diophantine analysis*, vol. 231 of *Grundlehren der mathematischen Wissenschaften.* Springer-Verlag, 1978.
9. 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.
10. Silverman, J. H. *The arithmetic of elliptic curves*, vol. 106 of *Graduate Texts in Mathematics.* Springer-Verlag, 1986.
11. 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.