Cryptanalysis of the Sakazaki-Okamoto-Mambo ID-based Key Distribution System over Elliptic Curves (Extended abstract)

Minghua Qu, Doug Stinson and Scott Vanstone
Certicom Corporation

Feb. 26, 2001

Abstract

In 1997, H. Sakazaki, E. Okamato and M. Mambo [4] proposed an ID-based key distribution system on an elliptic curve over $\mathbb{Z}_n$. We will cryptanalyze the scheme and demonstrate that when the hashed ID length is about 160 bits, the scheme is insecure. To be specific, after requesting a small number of keys from the Center, our attack allows a new valid key to be constructed without any further interaction with the Center.

1 Introduction

In 1986, E. Okamoto proposed an ID-based key distribution system (KDS) whose security depends on the difficulty of factoring a number of two large primes, as in the RSA public key cryptosystem. However, this scheme cannot be constructed on an elliptic curve over $\mathbb{Z}_n$ in a straightforward way because the point corresponding to a user's identity may not be defined on the elliptic curve. As a solution to this problem, Sakazaki-Okamoto-Mambo [4] proposed an ID-based KDS on an elliptic curve over $\mathbb{Z}_n$.The proposed scheme can be also constructed on the ring $\mathbb{Z}_n$.

We will show that some homomorphism-like properties hold in the Sakazaki-Okamoto-Mambo scheme, and use them to cryptanalyze the scheme. We will demonstrate that, when the hashed ID length is about 160 bits, one can forge a private key S_i corresponding to some identity I_i. Hence the Sakazaki-Okamoto-Mambo scheme is insecure.

This paper is organized as following: Section 2 describes the Sakazaki-Okamoto-Mambo KDS scheme. Section 3 will discuss the security of the scheme and present and analyze some attacks. Section 4 concludes the paper with some brief comments.

2 The Sakazaki-Okamoto-Mambo Scheme

2.1 Elliptic curves over $\mathbb{Z}_n$

For a detailed discussion of elliptic curves over $\mathbb{Z}_n$, see Koblitz [3]. Here we just provide enough information to describe the Sakazaki-Okamoto-Mambo scheme.

Let n be a product of two primes p and q. Let $a, b \in \mathbb{Z}_n$ be such that $\gcd(4a^3 + 27b^2, n) = 1$. An *elliptic curve* over $\mathbb{Z}_n$ with parameters a and b is defined as the set of points

$$\{(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_n : y^2 \equiv x^3 + ax + b \pmod{n}\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ is a special point called the *point at infinity* . This elliptic curve is denoted $E_n(a, b)$. Suppose that $G \in E_n(a, b)$ be a *base point* having order

$$k = \operatorname{lcm}(\#E_p(a, b), \#E_q(a, b)).$$

Note that $E_p(a, b)$ and $E_q(a, b)$ denote the corresponding elliptic curves defined over $\mathbb{Z}_p$ and $\mathbb{Z}_q$, and $\#E$ denotes the number of points in an elliptic curve E. Such a base point G exists provided that $E_p(a, b)$ and $E_q(a, b)$ are both cyclic groups.

2.2 The Sakazaki-Okamoto-Mambo Scheme over an Elliptic Curve

2.2.1 Set-up Phase

The Center publishes the parameters of an elliptic curve $E_n(a, b)$, and a base point G, as described in Section 2.1. The Center has private key consisting of k, p and q.

2.2.2 Issuing a Private Key to a User

Suppose the Center wants to issue a private key to a user i. Let $I_i = h(ID_i)$, where h is a public hash function, and ID_i is user i's public identifying information. We call I_i a *hashed identity*. Suppose that $\gcd(I_i, k) = 1$. The Center computes

$$D_i = I_i^{-1} \bmod k$$

and

$$S_i = -D_i G.$$

Hence, it follows that

$$I_i S_i + G = \mathcal{O}.$$

The Center transmits (I_i, S_i) to user i using a secure channel. S_i is user i's private key, and I_i is his public key.

2.2.3 Key Exchange Scheme

Suppose Alice and Bob want to establish a common key. Define $[1, n-1] = \{1, \ldots, n-1\}$. Let I_A, I_B be Alice's and Bob's public keys, and let S_A and S_B be their private keys.

First, Alice randomly chooses an integer $r_A \in [1, n-1]$, computes the elliptic curve point

$$C_A = S_A + r_A I_B G$$

over $E_n(a, b)$, and sends it to Bob. Similarly, Bob randomly chooses an integer $r_B \in [1, n-1]$, computes

$$C_B = S_B + r_B I_A G$$

over $E_n(a, b)$, and sends it to Alice.

Then Alice computes

$$K_{AB} = r_A(I_B C_B + G)$$

over $E_n(a, b)$, and Bob computes

$$K_{BA} = r_B(I_A C_A + G)$$

over $E_n(a, b)$. Obviously

$$K_{AB} = K_{BA} = r_A r_B I_A I_B G.$$

Note that the above scheme can also be described over $\mathbb{Z}_n$.

3 Cryptanalysis of the Sakazaki-Okamoto-Mambo Scheme

In this section, we will investigate a weakness of the Sakazaki-Okamoto-Mambo scheme. We will concentrate the private keys distributed by the Center. We will give methods to forge a private key S_I corresponding to a public key I, where I is a hashed identity.

3.1 Homomorphism-like Properties of the Sakazaki-Okamoto-Mambo Scheme

In the following definitions, we assume that the modulus k is unknown. All inverses are defined modulo k. For any positive integer x, define

$$S_x = -x^{-1}G.$$

(S_x is the private key corresponding to public key x.)

Lemma 3.1 *Let $z = xy$ where x, y and z are positive integers. Suppose that $S_z = -z^{-1}G$. Then $S_x = yS_z$ and $S_y = xS_z$.*

Proof. Clearly we have

$$xyz^{-1} \equiv 1 \pmod{k},$$

so it follows that

$$-x^{-1} \equiv -yz^{-1} \pmod{k}$$

and

$$-y^{-1} \equiv -xz^{-1} \pmod{k}.$$

Hence,

$$S_x = -x^{-1}G = -yz^{-1}G = yS_z$$

and

$$S_y = -y^{-1}G = -xz^{-1}G = xS_z.$$

$\square$

Lemma 3.2 *Suppose that* $\gcd(x, y) = 1$*, $Sx = -x^{-1}G$ and $S_y = -y^{-1}G$. Then $S_{xy} = k_1S_y + k_2S_x$, where k_1 and k_2 are integers that can be computed efficiently, given x and y.*

Proof. Since $\gcd(x, y) = 1$, the extended Eulcidean algorithm can be used to find integers k_1 and k_2 such that

$$k_1 x + k_2 y = 1.$$

It follows that

$$-(xy)^{-1} \equiv -k_1 y^{-1} - k_2 x^{-1} \pmod{k}.$$

Hence,

$$S_{xy} = -(xy)^{-1} G = -k_1 y^{-1} G - k_2 x^{-1} G = k_1 S_y + k_2 S_x.$$

□

3.2 Attacks on the Sakazaki-Okamoto-Mambo Scheme

Here is the basic idea of the attacks. If we know enough public keys I_i and their corresponding private keys S_i, then we can construct a database

$$DB := \{(x, S_x)\}$$

for small prime integers x, using Lemma 3.1. For a given public key I (i.e., a hashed identity), suppose that I can be factored as

$$I = x_1 x_2 \dots x_u,$$

where $\gcd(x_i, x_j) = 1$ for all $i \neq j$ and and $(x_i, S_{x_i}) \in DB$ for all i. Then we can compute the private key

$$S_I = -I^{-1} G$$

using Lemma 3.2.

We now present two attacks on the scheme that use this idea. The first attack is an attack on a specific pre-chosen identity. The second attack is more general, but less efficient. We will suppose that the length of a hashed identity, say I, is 160 bits. Let t be a positive integer. A positive integer m is *t-smooth* if all the prime divisors of m are less than t. (Typically we will choose $t = 2^{40}$.)

Algorithm 1 is a forgery of a private key S_I corresponding to a specific public key I (where I is the hash value of the identity information of a user i).

Algorithm 1

1. Find a t-smooth hashed identity $I = p_1 p_2 \ldots p_u$, where the p_i's are distinct primes.

2. Find a set of hashed identities $I_1, \ldots, I_v$ such that, for every i with $1 \leq i \leq u$, there exists an I_j with $1 \leq j \leq v$ such that $p_i | I_j$. (Clearly we can assume $v \leq u$.)

3. For every j with $1 \leq j \leq v$, obtain a private key S_{I_j} corresponding to public key I_j by interacting with the Center.

4. For every i with $1 \leq i \leq u$, compute S_{p_i} using Lemma 3.1.

5. Construct S_I from the u pairs (p_i, S_{p_i}) by repeated applying Lemma 3.2.

In Algorithm 1, we build a database that allows us to forge a specific secret key. Algorithm 2 consructs a large database that will allow various secret keys to be forged. More precisely, a secret key can be forged using Algorithm 2 for a hashed identity I whenever I is t-smooth and square-free.

Algorithm 2

1. Find a set of hashed identities $I_1, \ldots, I_w$ such that, for every prime $p < t$, there exists an I_j with $1 \leq j \leq w$ such that $p | I_v$.

2. For $1 \leq j \leq w$, obtain a private key S_{I_j} corresponding to public key I_j by interacting with the Center.

3. For all primes $p < t$, compute S_p using Lemma 3.1.

4. Let $I = p_1 p_2 \ldots p_u$ be a t-smooth hashed identity, where the p_i's are distinct primes.

5. Construct S_I from the u pairs (p_i, S_{p_i}) by repeated applying Lemma 3.2.

3.3 Analysis of the Complexity of the Attacks

In this section, we analyze the complexity of the attacks. First, we need some results on smoothness probabilities. Let $\Psi(x, t)$ denote the number of integers in the interval $[1, x]$ which are t-smooth. The notation "log " is used to denote a logarithm to the base e. The following result can be found in [1, p. 234].

Theorem 3.3 *For $x \geq 4$ and $2 \leq t \leq x$, it holds that $\Psi(x,t) > x^{1-\log\log x/\log t}$.*

If we take $t = x^\alpha$, where $0 < \alpha < 1/2$, then $\Psi(x,t) > x/(\log x)^{1/\alpha}$. Then the probability that a random integer in $[1,x]$ is t-smooth is at least $1/(\log x)^{1/\alpha}$. When $x = 2^{160}$ and $t = 2^{40}$, we have $\alpha = 1/4$, and the probability is at least

$$\frac{1}{(160\log 2)^4} = \frac{1}{1.5 \times 10^8} > \frac{1}{2^{28}}.$$

(In practice, however, the probability is much larger than this. In fact, when $1/2 \leq \alpha \leq 1$, the probability is close to $1 + \log \alpha$; see [2, p. 383].)

We first analyze Algorithm 1.

- Suppose we attempt to construct I in step 1 by choosing random identities, hashing them and testing them to see if they are t-smooth. We should find a suitable I after 2^{28} trials. Assuming that $I = p_1 p_2 \ldots p_u$ is square-free, we proceed to step 2.

- In step 2, we might choose random identities, hash them and test them for divisibility by the the p_i's. The probability that a random integer is divisible by p_i is $1/p_i$, so it will take about p_i trials to find a hashed identity divisible by p_i, for each i. The total number of trials will be about $p_1 + \ldots + p_u$. It is not hard to see that the number of trials is maximized when $u = 4$ and $p_1, p_2, p_3, p_4 \approx 2^{40}$. The number of trials in the worst case is therefore expected to be about $4 \times 2^{40} \approx 2^{42}$.

- In step 3, we require u interactions with the Center to obtain the S_{p_i}'s, $1 \leq i \leq u$. In the worst case, we will have $u = 30$, because the product of the first 31 primes exceeds 2^{160}.

- Finally, step 4 can be done quickly using $u - 1$ applications of Lemma 3.2.

In practice, the most time-consuming step is probably step 1. This is because the values I in step 1 need to be checked for divisibility by all the primes up to 2^{40}. In step 2, we are only testing for divisibility by the p_i's determined in step 1.

This attack is sufficient to cast doubt on the security of the Sakazaki-Okamoto-Mambo scheme if the length of a hashed identity is 160 bits.

Algorithm 2 can be analyzed in a similar fashion. Let $\pi(x)$ denote the number of primes that are less than x. (By the prime number theorem, $\pi(x) \approx x/\log x$.) Unfortunately, in step 2 of Algorithm 2, we need to construct a database of $\pi(2^{40})$ keys. This is so large that it is not really practical.

4 Summary

The attack presented in Algorithm 1 is at least close to being practical in the case where a hashed identity is 160 bits in length. After requesting a small number of (private) keys from the Center, our attack allows a new valid key to be constructed without any further interaction with the Center. This shows that it is not sufficient for the hash function to be "secure" in order for the Sakazaki-Okamoto-Mambo scheme to be secure.

References

[1] E. Bach and J. Shallit. *Algorithmic Number Theory. Volume 1: Efficient Algorithms*, MIT Press, 1996.

[2] D. Knuth. *The Art of Computer Programming, Volume 2, Seminumerical Algorithms (Third Edition)*, Addison-Wesley, 1998.

[3] N. Koblitz. *A Course in Number Theory and Cryptography (Second Edition)*, Springer-Verlag, 1994.

[4] H. Sakazaki, E. Okamato and M. Mambo. ID-based key distribution system over an elliptic curve, *Contemporary Mathematics* **225** (1999), 215–223 (Fourth International Conference on Finite Fields).