A Public Key Cryptosystem Based on Elliptic Curves over $\mathbb{Z}/n\mathbb{Z}$ Equivalent to Factoring

Bernd Meyer[1*] and Volker Müller[2]

[1] Universität des Saarlandes
Fachbereich Informatik
Postfach 15 11 50
66041 Saarbrücken
Germany
Email: bmeyer@cs.uni-sb.de

[2] Department of Combinatorics & Optimization
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1
Email: vmueller@crypto2.uwaterloo.ca

Abstract Elliptic curves over the ring $\mathbb{Z}/n\mathbb{Z}$ where n is the product of two large primes have first been proposed for public key cryptosystems in [4]. The security of this system is based on the integer factorization problem, but it is unknown whether breaking the system is equivalent to factoring. In this paper, we present a variant of this cryptosystem for which breaking the system is equivalent to factoring the modulus n. Moreover, we extend the ideas to get a signature scheme based on elliptic curves over $\mathbb{Z}/n\mathbb{Z}$.

1 Introduction

In recent years, elliptic curves over finite fields have gained a lot of attention. The use of elliptic curves over finite fields in public key cryptography was suggested by Koblitz [3] and Miller [7]. The security of these cryptosystems is based on the difficulty of the discrete logarithm problem in the group of points on an elliptic curve. Later Vanstone et. al. proposed to use elliptic curves over the ring $\mathbb{Z}/n\mathbb{Z}$, where n is the product of two large prime numbers [4]. The security of their public key cryptosystem is based on the factorization problem for n, but it is not known whether decryption is equivalent to factoring n. They use elliptic curves of special form such that the factorization of n directly gives the order of the group $E(\mathbb{Z}/n\mathbb{Z})$. The knowledge of the group order is important in the decryption procedure. A detailed description of all these systems can be found in [6].

[*] Author was supported by the Deutsche Forschungsgemeinschaft

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 49-59, 1996.

On the other hand, there exist several RSA-variants equivalent to factoring, see for example [10]. We use similar ideas to develop a new public key cryptosystem based on elliptic curves over the ring $\mathbb{Z}/n\mathbb{Z}$. For the new cryptosystem, decryption is equivalent to factoring n. Moreover, we discuss a generalization of our ideas to get a public key signature scheme using elliptic curves over $\mathbb{Z}/n\mathbb{Z}$.

The remainder of the paper is organized as follows. Section 2 gives a short introduction to elliptic curves over $\mathbb{Z}/n\mathbb{Z}$. In Sections 3 and 4, we describe the cryptosystem and the signature scheme. Section 5 describes some algorithms needed in the decryption part and proves the security of the system. Finally, Section 6 makes some concluding remarks.

2 Elliptic Curves over $\mathbb{Z}/n\mathbb{Z}$

In this section we will introduce basic facts about elliptic curves over the ring $\mathbb{Z}/n\mathbb{Z}$, where n is a square free rational integer not divisible by 2 and 3. A detailed description of the theory of elliptic curves can be found in [9].

Definition 1. Let $n \in \mathbb{N}$ be not divisible by 2 and 3. The projective plane $\mathcal{P}^2(\mathbb{Z}/n\mathbb{Z})$ is the set of all triples $(x, y, z) \in (\mathbb{Z}/n\mathbb{Z})^3 - \{(0,0,0)\}$ modulo the equivalence relation

$$(x, y, z) \sim (x', y', z') \Leftrightarrow \exists \lambda \in (\mathbb{Z}/n\mathbb{Z})^* \text{ with } x = \lambda\, x', y = \lambda\, y', z = \lambda\, z' \ .$$

Let $a, b \in \mathbb{Z}/n\mathbb{Z}$ with $4a^3 + 27b^2 \in (\mathbb{Z}/n\mathbb{Z})^*$. Then the set of points on the elliptic curve $E = (a, b)$ over $\mathbb{Z}/n\mathbb{Z}$ is defined as

$$E(\mathbb{Z}/n\mathbb{Z}) = \left\{(x : y : z) \in \mathcal{P}^2(\mathbb{Z}/n\mathbb{Z}) \mid y^2 z \equiv x^3 + axz^2 + bz^3 \bmod n\right\} \ .$$

The point $(0 : 1 : 0)$ is called *point at infinity.*

Let the prime factorization of n be given as

$$n = \prod_{i=1}^{k} p_i$$

where p_i are distinct prime numbers greater than 3. Using the Chinese Remainder Theorem it is easy to show that there is a bijection

$$E(\mathbb{Z}/n\mathbb{Z}) \cong E(\mathbb{Z}/p_1\mathbb{Z}) \times \ldots \times E(\mathbb{Z}/p_k\mathbb{Z}) \ . \qquad (1)$$

This bijection maps a point $(x : y : z) \in E(\mathbb{Z}/n\mathbb{Z})$ to a tuple of points

$$\Big((x \bmod p_1 : y \bmod p_1 : z \bmod p_1), \ldots, (x \bmod p_k : y \bmod p_k : z \bmod p_k)\Big) \ .$$

Note that $(x \bmod p_i : y \bmod p_i : z \bmod p_i)$ indeed is a point on $E(\mathbb{Z}/p_i\mathbb{Z})$ for all $1 \leq i \leq k$. It is well known that the set $E(\mathbb{Z}/p\mathbb{Z})$ (where p is a prime greater 3) has the structure of an abelian (usually additively written) group, where

$(0:1:0)$ is the zero element (see [9]). Using the bijection (1), we can define an addition on $E(\mathbb{Z}/n\mathbb{Z})$, such that $E(\mathbb{Z}/n\mathbb{Z})$ also forms an abelian group.

Usually, one has a slightly different notation for points on elliptic curves. Assume that the z-coordinate of the point $(x:y:z) \in E(\mathbb{Z}/n\mathbb{Z})$ is coprime to n. Then there exists a triple $(\tilde{x}, \tilde{y}, 1)$ in the set $(x:y:z)$. Such a point can be represented as a pair $(\tilde{x}, \tilde{y})$. On the other hand, points which can not be represented in this way directly lead to a factorization of n. It seems to be very unlikely to find such points, because the factorization problem is expected to be hard. Therefore we will represent points as a pair of a x- and y-coordinate.

With this representation, we can use exactly the same formulas for addition in $E(\mathbb{Z}/n\mathbb{Z})$ as given in [9] for fields. But note that there is a tiny probability that these formulas fail (which is equivalent to finding a factor of n). We will only describe the formulas for doubling points.

Let $E = (a,b) \in (\mathbb{Z}/n\mathbb{Z})^2$ be an elliptic curve and $P = (x,y) \in E(\mathbb{Z}/n\mathbb{Z})$ be a point with order greater two. Then we can double P, i.e. we can compute the point $2 \cdot P = (X,Y)$, using the formulas

$$\begin{aligned} \lambda &= (3x^2 + a) \cdot (2y)^{-1} , \\ X &= -2x + \lambda^2 , && (2) \\ Y &= -y + \lambda\,(x - X) . && (3) \end{aligned}$$

3 The Public Key Cryptosystem

In this section we will describe the new public key cryptosystem based on elliptic curves over $\mathbb{Z}/n\mathbb{Z}$. The security of this system is based on the integer factorization problem, as will be shown later.

Assume that n is the product of two large primes p, q, where $p, q \equiv 11 \bmod 12$. As remarked in [10], each square in $(\mathbb{Z}/n\mathbb{Z})^*$ has exactly four square roots of whom exactly one is itself a square (especially, n is a Blum integer). Moreover, for every element in $(\mathbb{Z}/n\mathbb{Z})^*$ there exists exactly one cube root. We can classify the square roots of a square κ even further: exactly two square roots of κ have Jacobi symbol $+1$ (so called type I roots), the two others have Jacobi symbol -1 (type II roots). If a is a type I root of a square κ (resp. type II root), then $-a$ is the other type I root (resp. type II root). Moreover we can distinguish between a and $-a$: consider elements in $(\mathbb{Z}/n\mathbb{Z})^*$ as numbers in the set $\{1, \ldots, n-1\}$. Then exactly one of the two elements a and $-a$ is an even number (note that n is an odd number). For $a \in (\mathbb{Z}/n\mathbb{Z})^*$, we define $\mathrm{lsb}(a)$ to be the least significant bit of the "number" a. Thus, given a square $\kappa \in (\mathbb{Z}/n\mathbb{Z})^*$, we can identify each square root of κ with the help of two bits, the type and the least significant bit.

The keys of the cryptosystem are then given in the following way: The public key of Bob is the modulus n. The two prime factors p and q of n form the private key of Bob and are kept secret.

Assume that Alice wants to send a "message" m to Bob, where $0 < m < n$ (in the following we always assume that text messages are transformed into numbers in some publicly known way). She encrypts this message m with the help of Bob's public key in the following way:

Encryption procedure:

```
(1)  choose $\lambda \in \mathbb{Z}/n\mathbb{Z} - \{0\}$ at random.
(2)  set $P = (m^2,\ \lambda\, m^3)$.
(3)  set $a = \lambda^3$ and compute $b = (\lambda^2 - 1)\, m^6 - a\, m^2$.
(4)  if ($\gcd(4a^3 + 27b^2, n) > 1$) then
(5)     return (protocol error: $4a^3+27b^2 \equiv 0 \bmod n$ or factor of $n$ found)
(6)  fi
(7)  send $E = (a, b)$, $x(2 \cdot P)$, type($y(2 \cdot P)$) and lsb($y(2 \cdot P)$) to Bob.
```

Note that the computation of the gcd in step (4) can be omitted, if one accepts that the probability of "guessing" a factor of a large integer n is extremely small. Then the encryption procedure can be done with only a few multiplications and one inversion modulo n (in the doubling part of P).

Assume that Bob receives an encrypted message E, x_Q, t, l, where $E = (a, b)$ is an elliptic curve over $\mathbb{Z}/n\mathbb{Z}$, $x_Q \in \mathbb{Z}/n\mathbb{Z}$ and t, l are two bits. He can decrypt this message using the following decryption procedure:

Decryption procedure:

```
(1)  compute the square root $y_Q$ of $x_Q^3 + a\, x_Q + b$ with type $t$ and lsb $l$.
(2)  set $Q = (x_Q, y_Q)$.
(3)  compute all points $P_i \in E(\mathbb{Z}/n\mathbb{Z})$, $1 \le i \le s$, such that $2 \cdot P_i = Q$.
(4)  compute set $I = \{1 \le i \le s;\ a^2 = y(P_i)^6\, x(P_i)^{-9}\}$.
(5)  if ($\#I > 1$) then
(6)     return (protocol error: more than one solution)
(7)  else
(8)     return ($m = y(P_I)^3\, x(P_I)^{-4}\, a^{-1}$)
```

Note that in step (8) the index set I must have only one element, such that the notation P_I denotes the point P with the index given in I. The correctness of the decryption algorithm follows directly since the "correct plain text point" P satisfies both tests in step (3) and (4).

The coefficient a of the elliptic curve E determines λ exactly. However, the determination of λ given only the ciphertext is supposed to be difficult if the factorization of n is unknown. If an intruder would know λ, then the cryptosystem would be equivalent to the cryptosystem of Williams [10], as was pointed out to us by M. Joye and J.-J. Quisquater [1].

In the following theorem, we will consider the case that the index set has more than one element.

Theorem 2. *The probability that the index set I in step (4) of the decryption procedure has more than one element and we cannot factor the modulus n is at most* $118^2/(n-1)$.

Proof. We have shown that the input to the decryption procedure uniquely determines the point Q on the elliptic curve E. By construction, there is a point P on E ("the embedded message") which satisfies the conditions in step (3) and (4) of the decryption procedure. Then we can express the coefficients of E and $x(Q)$ as rational functions in $x(P)$ and λ (in the following we will consider the coefficients of all polynomials as rational functions in λ and $x(P)$). In Lemma 3 we will show that there exists a degree 4-polynomial $f(X) \in (\mathbb{Z}/n\mathbb{Z})(\lambda, x(P))[X]$ such that x-coordinates of points passing test (3) are zeros of f. Passing step (4) means that x-coordinates of such points have to be zeros of a degree 9-polynomial $g(X)$.

Assume that there exists a point P_2 different from P which passes both tests. Then $x(P_2)$ has to be a zero of the remainder $k(X)$ of $g(X)/(X - x(P))$ and $f(X)/(X - x(P))$, which is a degree 2-polynomial (or we find a non trivial factor of n). On the other hand, we know that P_2 is the sum of P and a two-torsion point $(r, 0) \in E(\mathbb{Z}/n\mathbb{Z})$. We express the x-coordinate of P_2 as a rational function in $x(P)$ and r and substitute this expression into $k(X)$. Using the fact that $r^3 + a\,r + b \equiv 0 \bmod n$, we obtain an equation in $x(P)$ and λ which must be zero. This equation has "degree" 118 in λ. Therefore, for given and fixed $x(P)$, there exist at most 118^2 different values for λ such that this equation vanishes modulo n. Since we choose λ at random in step (1) of the encryption procedure, the probability of choosing a "bad" λ is at most $118^2/(n-1)$. □

Since the modulus n has to be a number difficult to factor (i.e. it should be sufficiently large), Theorem 2 shows that the error probability for the decryption procedure is very small.

We have not yet mentioned how to solve the so called *square root problem in* $E(\mathbb{Z}/n\mathbb{Z})$ in step (3). We will explain an algorithm for doing this in Section 5. In addition, we will give an upper bound for the running time of the encryption and decryption procedure in Theorem 5. In the following section we will explain how the idea of using square roots of points on elliptic curves over $\mathbb{Z}/n\mathbb{Z}$ can be used to construct a public key signature scheme.

4 A Signature Scheme

In this section we extend the ideas given in the last section to describe a public key signature scheme. Assume that the public key and private key of Bob are chosen as in the last section, i.e. Bob's public key is an integer n which is the product of two primes $p, q \equiv 11 \bmod 12$ and his secret key is the knowledge of the two prime factors p, q of n. We assume that some publicly known cryptographic hash function is used to compute fingerprints for messages.

Signing a message m:

```
(1)  compute a fingerprint $0 \leq m' < n$ for the message $m$ using a cryptographic hash function.
(2)  find $\lambda$, $\zeta \in (\mathbb{Z}/n\mathbb{Z}^*)$ such that for the elliptic curve $E = (\lambda^3, .)$ we
     get $x(2 \cdot (\zeta^2, \lambda\zeta^3)) = m'$ and $y(2 \cdot (\zeta^2, \lambda\zeta^3))$ has type I.
(3)  return (Signature for $m'$ is $(\zeta, \lambda)$)
```

Note that the formulas for doubling a point do not depend on the coefficient b of the given elliptic curve. As will be shown in the next section, the problem in step (2) can be solved by finding a root of a polynomial in the two variables λ and ζ. Since the factorization of n is known to Bob, he solves this problem modulo p (resp. q) and uses the Chinese Remainder Theorem to compute the values for λ, ζ modulo n. The checking of a signature is then obvious:

Checking a signature:

```
(1)  set $a = \lambda^3$ and $P = (\zeta^2, \lambda\zeta^3)$.
(2)  compute $Q = 2 \cdot P$ on $E = (a, (\lambda^2 - 1)\zeta^6 - a\zeta^2)$.
(3)  check whether $x(Q) = m'$ and $y(Q)$ has type I.
(4)  check whether $m'$ is a correct fingerprint for $m$.
(5)  if (all tests are successful) then
(6)      return (accept the signature.)
(7)  else
(8)      return (reject the signature.)
```

It should be mentioned that Bob should be very careful in signing arbitrary messages. As will be shown in Theorem 6, this signature scheme is very vulnerable to a chosen plaintext attack where a hash value m' is given to Bob for signing.

In the next section, we will consider the square root problems for various situations. The interesting cases are elliptic curves over prime fields and over $\mathbb{Z}/n\mathbb{Z}$, where the factorization of n is either known or unknown. We will show that the ability of signing arbitrary messages is equivalent to knowing the factorization of n.

5 The Square Root Problem in $E(\mathbb{Z}/n\mathbb{Z})$

In this section, we will consider the missing parts in the decryption algorithm. Obviously, decryption can be done if we can invert the operation of doubling a point. Therefore we study the so called *square root problem*, which has the following form:

Given a point $Q \in E(\mathbb{Z}/n\mathbb{Z})$. Compute all points $P \in E(\mathbb{Z}/n\mathbb{Z})$ with

$$2 \cdot P = Q \ .$$

Such points P will be called ***square roots*** of Q. Note that the notation "square root" is used to show the analogy to the corresponding problem for the ring $(\mathbb{Z}/n\mathbb{Z})^*$. In the following subsection, we describe a solution to this problem if n is a prime number.

5.1 Computing Square Roots in $E(\mathbb{Z}/p\mathbb{Z})$

Let E be an elliptic curve defined over a prime field $\mathbb{Z}/p\mathbb{Z}$, where p is a prime number greater than 3 and let Q be any point on E for which we want to solve the square root problem. First note that the existence of a square root is not necessarily assured. In contrary, there exist points Q such that there is no square root of Q. An easy example for such a situation occurs, if E is a cyclic group of even order and Q is a generator of $E(\mathbb{Z}/p\mathbb{Z})$. Nevertheless there is an algorithm for solving this existence problem and the square root problem for Q. The algorithm is based on the following lemma:

Lemma 3. *Let $E = (a,b) \in (\mathbb{Z}/p\mathbb{Z})^2$ be an elliptic curve over $\mathbb{Z}/p\mathbb{Z}$ and $Q = (x_Q, y_Q) \in E(\mathbb{Z}/p\mathbb{Z})$, $Q \neq \mathcal{O}$. If $P \in E(\mathbb{Z}/p\mathbb{Z})$ is a square root of Q, then the x-coordinate of P is a root (in $\mathbb{Z}/p\mathbb{Z}$) of the polynomial $f(X,a,b,x_Q) \in (\mathbb{Z}/p\mathbb{Z})[X]$, where*

$$f(X,a,b,x_Q) \ = \ X^4 - 4\,x_Q\,X^3 - 2\,a\,X^2 - (4\,a\,x_Q + 8\,b)\,X + a^2 - 4\,b\,x_Q \ .$$

Proof. Using the doubling formula (2), we can compute the x-coordinate of $2 \cdot (X,Y)$ for an arbitrary point (X,Y) of order greater 2 as

$$-2 \cdot X \quad + \quad \left(\frac{3\,X^2 + a}{2\,Y}\right)^2 \ .$$

Equalizing this with the x-coordinate of Q and using simple transformations, we obtain the result of the lemma. □

This lemma is the basis for the following algorithm which solves the square root problem in $\mathbb{Z}/p\mathbb{Z}$. First we check whether the polynomial given in the lemma has a root in $\mathbb{Z}/p\mathbb{Z}$. If there is no root, then there cannot exist a square root of Q and we exit. Otherwise we compute all roots in $\mathbb{Z}/p\mathbb{Z}$ of the polynomial $f(X,a,b,x_Q)$, where a, b, x_Q are given as input. One should observe that not every root $x_P \in \mathbb{Z}/p\mathbb{Z}$ of $f(X,a,b,x_Q)$ is actually a x-coordinate of a square root of Q. It might be that $x_P^3 + a x_P + b$ is not a square in $\mathbb{Z}/p\mathbb{Z}$ and that x_P is a x-coordinate of a point in the quadratic extension of $\mathbb{Z}/p\mathbb{Z}$. In our context, we are only interested in square roots defined over the prime field such that the second step of an algorithm has to check whether a root x_P actually is the x-coordinate of a point defined over $\mathbb{Z}/p\mathbb{Z}$. In addition, the x-coordinate of a point does not specify the point exactly, since P and $-P$ have the same x-coordinate. Therefore we have to check whether $2 \cdot P = Q$ or whether $2 \cdot (-P) = Q$. These observations lead to the following Algorithm 4 which solves the square root problem for prime fields $\mathbb{Z}/p\mathbb{Z}$.

Algorithm 4.

```
INPUT: Elliptic curve $E = (a, b) \in (\mathbb{Z}/p\mathbb{Z})^2$, $Q \in E(\mathbb{Z}/p\mathbb{Z})$.
OUTPUT: Set $S$ of all square roots of $Q$ in $E(\mathbb{Z}/p\mathbb{Z})$.

(1)  $S = \emptyset$.
(2)  compute all roots of the polynomial $f(X, a, b, x(Q))$ in $\mathbb{Z}/p\mathbb{Z}$.
(3)  for (all roots $x_P$ computed in step (2)) do
(4)    if ($x_P^3 + a\,x_P + b$ is a square in $\mathbb{Z}/p\mathbb{Z}$) then
(5)      compute a square root $y_P$ of $x_P^3 + a\,x_P + b \bmod p$.
(6)      check whether $2\cdot(x_P, y_P) = Q$ or $2\cdot(x_P, -y_P) = Q$, add a found
         square root to $S$.
(7)    fi
(8)  od
(9)  return ($S$)
```

All steps of Algorithm 4 can be done using a factorization routine for polynomials over $\mathbb{Z}/p\mathbb{Z}$. There exists a probabilistic algorithm which computes all roots of a fixed degree polynomial modulo a prime p in expected time $O(\log(p)^3)$ (for a detailed treatment of polynomial factorization see e.g. [8]). Thus the expected running time of Algorithm 4 is $O(\log(p)^3)$.

5.2 Computing Square Roots in $E(\mathbb{Z}/n\mathbb{Z})$

Next we discuss the square root problem for elliptic curves over a ring $\mathbb{Z}/n\mathbb{Z}$ where n is a composite number. We distinguish two situations:

Factorization of n Known Let us first consider the problem when the factorization of n is known, e.g.

$$n = \prod_{i=1}^{k} p_i$$

and $p_i \in \mathbb{P}_{>3}$. Using the isomorphism map (1) defined in Section 2, the square root problem in $E(\mathbb{Z}/n\mathbb{Z})$ can be reduced to the solution of k many square root problems in $E(\mathbb{Z}/p_i\mathbb{Z})$, $1 \leq i \leq k$.

Assume that we want to compute all square roots of a point $Q \in E(\mathbb{Z}/n\mathbb{Z})$. For all prime factors p_i of n we proceed in the following way: first reduce the x-

and y-coordinate of Q modulo p_i and obtain $Q_{p_i} \in E(\mathbb{Z}/p_i\mathbb{Z})$. Then we compute all square roots of the reduced point Q_{p_i} in $E(\mathbb{Z}/p_i\mathbb{Z})$ with Algorithm 4. If there exists no square root of Q_{p_i}, we have proven that there cannot exist a square root of Q in $E(\mathbb{Z}/n\mathbb{Z})$ and we return. Otherwise we know sets S_i of all square roots of Q_{p_i} for all $1 \leq i \leq k$. Since the map (1) is an isomorphism, we can compute all square roots of Q in $E(\mathbb{Z}/n\mathbb{Z})$ by using the Chinese Remainder Theorem for all elements of $S = S_1 \times \ldots \times S_k$. Clearly, this shows that the number of square roots of Q in $E(\mathbb{Z}/n\mathbb{Z})$ is exactly the product of the cardinalities of the sets $S_i, 1 \leq i \leq k$. Obviously, two square roots can only differ by a point of order 2. It is well known (see [9]) that there are at most 4 points of exponent 2 in $E(\mathbb{Z}/p_i\mathbb{Z})$ such that $\#S_i \leq 4$. Therefore the square root problem with given factorization of the modulus n can be solved in probabilistic time $O(4^k \log(n)^3)$, where k is the number of prime factors of the modulus n.

It is well known that the extended euclidean algorithm for computing inverses in $(\mathbb{Z}/n\mathbb{Z})^*$ needs time $O(\log(n)^2)$. Using these observations in the special situation of the cryptosystem presented in Section 3, we can derive the following running time result.

Theorem 5. *The encryption procedure takes time $O(\log(n)^2)$, decryption can be done in probabilistic time $O(\log(n)^3)$.*

Factorization of n Unknown The security of the cryptosystem presented in Section 3 is based on the intractability of solving the square root problem in $E(\mathbb{Z}/n\mathbb{Z})$ when the factorization of n is not known. In this section we will show that the existence of an algorithm for decrypting encrypted messages without knowledge of the factorization of the modulus n would induce a probabilistic polynomial time algorithm for factoring n.

Theorem 6. *Assume there is an oracle which can decrypt encrypted messages. Then there exists a probabilistic polynomial time algorithm for factoring n.*

Proof. We use the help of the oracle to compute two square roots of different type in $(\mathbb{Z}/n\mathbb{Z})^*$. This will give us a non trivial factor of n. We proceed as follows:

1. choose elements κ, $\lambda \in (\mathbb{Z}/n\mathbb{Z})^*$ at random.
2. compute $a = \lambda^3$ and $b = (\lambda^2 - 1)\,\kappa^6 - a\,\kappa^2$.
3. compute $x_Q = \left((9 - 8\,\lambda^2)\,\kappa^8 + 6\,\kappa^4\lambda^3 + \lambda^6\right)\left(4\,\lambda^2\kappa^6\right)^{-1}$ and
$$t = \text{type}\Big(2\,\lambda\,\left((36\,\lambda^2 - 27 - 8\,\lambda^4)\,\kappa^{12} + (12\,\lambda^2 - 27)\,\lambda^3\kappa^8 - 9\,\kappa^4\lambda^6 - \lambda^9\right)\Big)\,.$$
4. If type$(\kappa) = t$, then call the oracle with input $E = (a, b)$, x_Q, type II and arbitrary lsb; otherwise use the input $E = (a, b)$, x_Q, type I, arbitrary lsb. The oracle outputs a message m.
5. compute a non trivial divisor of n as $\gcd(\kappa - m, n)$.

We have to show the correctness of this procedure. Assume that $m \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $m^2 = \kappa^2$ and set $P = (m^2, \lambda\, m^3)$. Then P is a point on the elliptic curve $E = (a, b)$ computed in step 2. We compute the x-coordinate of $2 \cdot P$ as

$$x(2 \cdot P) \quad = \quad \frac{(9 - 8\,\lambda^2)\, m^8 + 6\, m^4 \lambda^3 + \lambda^6}{4\,\lambda^2 m^6}\ .$$

Since $\kappa^2 = m^2$, the value x_Q computed in step 3 is equal to $x(2 \cdot P)$. If we compute the Jacobi symbol of $y(2 \cdot P)$ over n, we obtain

$$\left(\frac{m}{n}\right) \left(\frac{2\,\lambda\, \left((36\,\lambda^2 - 27 - 8\,\lambda^4)\, m^{12} + (12\,\lambda^2 - 27)\,\lambda^3 m^8 - 9\, m^4 \lambda^6 - \lambda^9\right)}{n}\right)\ .$$

Therefore the choice of the type in step 4 of the above procedure guarantees that the oracle indeed "computes" a message m whose type is different from type(κ). Hence, we know two square roots of κ^2 of different types (namely κ and m), which surely factors n and we are done. □

This theorem shows that the ability to decrypt an arbitrary message without knowing the factorization of the public key n is equivalent to factoring n. Since the factorization problem for integers is believed to be hard, it should not be possible to decrypt arbitrary messages without the secret key. Exactly the same arguments as given in the proof of Theorem 6 can be used to show that the ability of signing arbitrary messages is equivalent to factoring n.

In the last section of this paper, we will discuss some practical aspects of the described cryptosystem.

6 Advantages and Disadvantages

The cryptosystem presented in this paper is a generalization of the system presented in [4]. Using small "encryption exponents" as 2 in our system has the advantage of speeding up the encryption process. On the other hand, our decryption process seems to be a bit more difficult than the decryption process of [4], but usage of known root formulas for polynomials of degree 4 might speed up our decryption procedure. Therefore, a comparison of the two systems can only be done by actually implementing and comparing the two systems in practice. An implementation of our system will probably be done in future.

One disadvantage versus [4] is obviously the fact that for the system of this paper one has to develop different code for encryption and decryption, whereas in [4] en- and decryption can be done by a fast exponentiation.

A disadvantage of all cryptosystems equivalent to factoring is the fact, that such systems are vulnerable to chosen ciphertext attacks. This was already mentioned in the paper of Williams [10]. A chosen ciphertext attack can simulate the proof of Theorem 6 and so factor the public key n. There is no way to come around this problem.

Another attack on elliptic curve cryptosystems is the so called low exponent attack (see [5]). This attack shows that one should not send the same message to more than 11 people. If an eavesdropper knows the 11 encrypted messages, he can find the original message in polynomial time. This attack is very similar to the low exponent attack on RSA. One referee proposed to append a few bits (for example the user name) to a message before encrypting and sending it. At the moment, we do not know whether this proposal will make the low exponent attack useless. This question has to be examined by further research.

Acknowledgment: We are very grateful to Marc Joye and Jean-Jacques Quisquater who pointed out to us the equivalence of an earlier version of the cryptosystem of this paper to the cryptosystem of Williams. (See [2, 10].) In this earlier version, the parameter λ in step (2) of the encryption procedure was a fixed square root of 2 which was part of Bob's public key.

References

1. M. Joye, J.-J. Quisquater: *Personal communication*
2. M. Joye, J.-J. Quisquater: *Note on the public-key cryptosystem of Meyer and Müller*, Technical Report CG-1996/2, Université catholique de Louvain
3. N. Koblitz: *Elliptic curve cryptosystems*, Mathematics of Computation, **48** (1987), 203–209
4. K. Koyama, U. Maurer, T. Okamoto, S. Vanstone: *New Public-Key Schemes Based on Elliptic Curves over the Ring $\mathbb{Z}_n$*, Advances in Cryptology: Proceedings of Crypto '91, Lecture Notes in Computer Science, **576** (1991), Springer-Verlag, 252–266
5. K. Kurosawa, K. Okada, S. Tsujii: *Low exponent attack against elliptic curve RSA*, Advances in Cryptology–ASIACRYPT '94, Lecture Notes in Computer Science, **917** (1995), Springer-Verlag, 376–383
6. A. Menezes: *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers (1993)
7. V. Miller: *Uses of elliptic curves in cryptography*, Advances in Cryptology: Proceedings of Crypto '85, Lecture Notes in Computer Science, **218** (1986), Springer-Verlag, 417–426
8. V. Shoup: *A New Polynomial Factorization Algorithm and its Implementation*, Preprint, (1995)
9. J. H. Silverman: *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, **106**, Springer-Verlag, (1986)
10. H.C. Williams: *A modification of the RSA public-key encryption procedure*, IEEE Transactions on Information Theory, **IT-26**, No. 6, (1980), 726–729