Security of elliptic curve cryptosystems over $\mathbb{Z}_n$

Ruoxi Hu[1, 4], **Weihong Wu**[2, 3, 5]

[1] The Vanguard school, 1605 S. Corona Avenue, Colorado Springs, CO,USA
[2] University of California, Riverside ,USA
[3] Corresponding author

[4] Sissi.hu@hotmail.com
[5] embarkwu@gmail.com

Abstract. Elliptic curves over Galois fields are widely used in modern cryptography. Cryptosystems based on elliptic curves are commonly deemed more secure than RSA for a given key size. However, with the rapid progress of quantum computing, the security of this traditional systems faces unprecedented challenge. To address this concern, this paper explores the resilience of a generalization of traditional elliptic curve cryptography. That is, we explore elliptic curves over non-prime rings (Zn), instead of fields. Elliptic curves over Zn for a composite integer n has been considered by researchers on information security. However, it is unclear how they stand against the unparalleled power of quantum computers. This article investigates quantum attacks on cryptosystems based on this new paradigm. The conclusion sheds light on the pressing and important task of searching for post-quantum cryptographic systems. In particular, the effectiveness of Shor's algorithm (or its variation) on such systems is analyzed.

Keywords: elliptic curve, cryptosystems, security.

1. Introduction

Cryptography is one of the most important applications of number theory and is used today to protect our private information. Elliptic curve cryptography is a widely used system among the various methods. It has emerged as one of the most promising cryptographic systems in modern security industry due to its strength and efficiency [1-5]. It is especially favored for its ability to provide the same level of security as traditional systems, such as RSA, with much smaller key sizes [2].

However, with the looming quantum computing breakthrough, modern cryptography faces a major challenge [6, 7]. Quantum computers, based on the principle of quantum physics, possess an unparalleled computing power that can break many classical cryptographic schemes upon which our daily life relies. Unfortunately, among these schemes, the usual elliptic curve cryptography is no exception. The widely celebrated work of Shor has made it possible to solve certain underlying mathematical puzzle which protects the scheme from classical computer attacks.

However, not all hopes are lost. Conventionally, elliptic curves used in elliptic curve cryptosystems are defined over prime fields, such as the integer modulo a prime number Zp. Never the less, recent research has shown a growing interest in exploring elliptic curves defined over non-prime rings, specifically Zn with n not a prime number. The reason behind this exploration is both theoretical and practical. Theoretically, it poses new mathematical challenges for mathematicians and computer scientists. Practically, they allow better felxibility in terms of finding desirable curves.

This study delves into the quantum attacks targeting at elliptic curves over non-prime rings. Our research sheds light on the vulnerability and security implications in such a scenario. The key objectives of this article are as follows:

To provide a user-friendly guide to classical cryptographic systems in general, and elliptic curves over non-prime rings in particular. This includes the mathematical background and construction framework.

To analyze the current state of quantum computing and its attacks on these cryptographic systems. It focus on Shor's algorithm and its effectiveness on elliptic curves.

To assess the potential risks posed by quantum attacks specifically targeted at elliptic curves over non-prime rings. The focus is on their advantages, limitations, and possible mitigation.

2. Overview of public-key cryptography

The elliptic curve cryptosystem concerning this article is an example of public key cryptography. In public key cryptography, the sender, Alice, wants to send the recipient, Bob, a message over an unsafe channel, meaning that anyone can access the information they send to each other. Both Alice and Bob choose private keys which they encrypt and send to each other over the unsafe channel. Then they both use their private key and the encrypted keys they have received and get to the key K which they can now use to encrypt messages. The security of these systems is based on one-way functions, which is one where given x it is easy to calculate y, but given y, it is extremely difficult to find x. Examples of these are factoring and the discrete logarithm problem, which is calculating the logarithm of a number mod n. Multiplication and exponents over a finite field are easy to calculate, but there is no known algorithm to efficiently factor an extremely large number or find the logarithm of a number over a finite field. The following section outlines an algorithm to efficiently find high powers of a number.

2.1. Binary exponentiation

Powers are calculated by repeatedly multiplying the number by itself, and the time complexity to calculate this is about $O(n \cdot \log^2 p)$. However, this can take lots of time when the power is large. Therefore, to quickly calculate the number $a^n \bmod p$ when n is large, one can first convert n to base 2. For example, when $n = 25$, the binary representation of n would be 11001_2. Therefore, one can represent a^n as

$$a^{11001_2} = a^{1_2} \cdot a^{1000_2} \cdot a^{10000_2} \tag{1}$$

one can compute this using an algorithm that starts with $t = a^0 = 1$ and repeatedly squares t. If the ith digit of the base 2 representation of n is 1, one multiply the answer by t. Then one square t and repeat the process until the power reaches n. The time complexity to calculate p^n using this method is only about $O(\log n \cdot \log^2 p)$ which is significantly faster than $O(n \cdot \log^2 p)$, and the final answer is a^n mod p. One can also see that when n is small, this optimization barely makes a difference in the time needed to calculate the power since the $O(n)$ complexity itself takes very little time. Therefore, to make the reverse problem of factoring or discrete logarithms harder, one needs to use very large numbers as parameters.

2.2. Selecting large primes

Public key cryptosystems often use large primes as their parameters for reasons stated above. Therefore, in order to find large primes, one can use Fermat's little theorem which states that for all primes p, any number $n^{p-1} \equiv 1 \bmod p$. One can test if a number is prime by selecting any random number a and calculating $a^{p-1} \bmod p$, and if the result is not 1, then p is not prime. When selecting a large prime several thousand bits long, one can simply randomly choose numbers of that length and test them until one find a number that is prime.

2.3. *Cryptography*

With the algorithms established above, there are two public key cryptosystems that are commonly used.

2.3.1. Diffie-Hellman key exchange. In the Diffie-Hellman key exchange, the sender Alice wants to send a message to Bob. They both pick a very large prime p and share this with each other over the unsafe channel. The prime p is public which means that anyone can have access to it. They then choose another number g between 1 and $p- 1$ and share g over the unsafe channel as well. The numbers p and g are the parameters to this scheme.

Then, Alice picks another value a where $0 \leq a \leq p - 2$. She keeps this value a secret and computes $A \equiv g^a \bmod p$. Meanwhile, Bob does also choose a value b in the range 0 to $p - 2$, keeps it secret, and calculates $B = g^b \bmod p$. Then, Alice and Bob each send their values A and B over the unsafe channel, keeping a and b secret.

Now, Alice can compute a result

$$S = B^a \, mod \, p \tag{2}$$

and Bob can also compute

$$S = A^a \, mod \, p \tag{3}$$

Since

$$S = (g^a)^b \, mod \, p = \left(g^b\right)^a mod \, p = g^{ab} \, mod \, p \tag{4}$$

Alice and Bob end up with the same result that no one else knows. They can then use this value S as their shared key to further encrypt and decrypt messages. Although their conversation is entirely public, eavesdroppers are unable to get to $S = g^{ab} \bmod p$ from only g^a and g^b while both Alice and Bob learn this value S. The security of this key exchange relies on the discrete logarithm problem discussed earlier. It is easy for Alice to calculate g^a and g^{ab} using the exponentiation algorithm but extremely hard for an eavesdropper to calculate a and b from only knowing $g^a \bmod p$ and $g^b \bmod p$.

2.3.2. El Gamal encryption. In El Gamal encryption, the parameters are also a large prime p and a number g satisfying $1 < g < p - 1$. Like in Diffie Hellman key exchange, Alice and Bob first agree on the large prime and number p and g. Bob then chooses a number b satisfying $0 \leq b \leq p - 2$ and calculates $B = g^b \bmod p$. Bob then publishes B and keeps b secret, making B a public key and b a private key.

Then, Alice will choose a random number r from 0 to $p - 2$. She will send a ciphertext to Bob for her message m in the range 0 to $p - 1$ by calculating the two numbers

$$R = g^r \, mod \, p \tag{5}$$

$$S = m \times B^r \, mod \, p \tag{6}$$

She forms a pair using these two numbers and sends Bob the ciphertext (R, S).

When Bob receives the pair, he computes the number

$$R^{-b} \times S = (g^r)^{-b} \times (m \times B^r) = g^{-rb} \times m \times g^{br} = m \, mod \, p \tag{7}$$

and therefore gets to the original message m sent by Alice. This is basically a Diffie-Hellman key exchange with the shared key being $K = g^{rb} = B^r = R^b$. Bob uses the public key B from the beginning and Alice uses the new public key $R = g^r \bmod p$ which is chosen at random and used only for this instance.

2.4. *The catch*

There are still some problems with these algorithms. Although Bob can simply publish his public key to everyone, a third person could also pretend to be Bob and publish their own public key. They can then intercept all Alice's messages to Bob that she encrypts using this public key. Therefore, in order to have a secure conversation with these public-key methods, Alice needs a channel that ensures the integrity of Bob's public key. This means that it does not need to be able to keep the key secret from everyone, but it needs to ensure that the key does indeed come from Bob and not someone posing as Bob. Most methods to obtain these channels involve Alice and Bob meeting in person in advance to exchange the key.

3. Elliptic curves

The cryptosystems this article focuses on are based on elliptic curves over finite fields. This section defines what elliptic curves are and introduce some operations used in the cryptosystems.

3.1. Elliptic curves over a finite field

An elliptic curve over a field K with the parameters $a, b \in K$ that satisfy $4a^3 + 27b^2 \neq 0$ is defined as the point at infinity O together with the set of all points (x, y) with $x, y \in K$ that satisfy

$$y^2 = x^3 + ax + b \tag{8}$$

The elliptic curve with parameters a, b and over the finite field F_p with p elements will be denoted as $E_p(a, b)$.

Addition of two points on an elliptic curve is computed by drawing a line through the two points, and the result is the third intersection between this line and the curve. For two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on an elliptic curve E, addition of P and Q is defined as the following. If P is equal to the neutral element O, then $P + Q = Q$ by definition of the neutral point O. If $x_1 = x_2$ and $y_1 = -y_2$, then $P + Q = O$. In all other cases, $P + Q$ can be computed as follows. Let λ be defined as

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \; if \; x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} \; if \; x_1 = x_2 \end{cases} \tag{9}$$

When $P + Q \neq O$, the denominator is always nonzero and λ is always defined. The point $P + Q = (x_3, y_3)$ is defined by

$$x_3 = \lambda^2 - x_1 - x_2 \tag{10}$$

$$y_3 = \lambda(x_1 - x_3 - y_1) \tag{11}$$

When the field is F_p, the curve does not actually look like a curve anymore. However, addition is still computed the same way modulo p.

There are also some lemmas that come with elliptic curves over a finite field. For an elliptic curve $E_p(a, b)$, let $\#E_p(a, b)$ denote the order (number of elements) of $E_p(a, b)$. From the Hasse bound it is known that $\#E_p(a, b) = p + 1 + t$ where t is between $-2\sqrt{p}$ and $2\sqrt{p}$. Schoof's algorithm provides a polynomial time solution to find the order of an elliptic curve, but this algorithm is impractical for large values of p. However, one can easily find the order of some special elliptic curves. An elliptic curve $E_p(a, b)$ is either a cyclic group or the product of two cyclic groups $\mathbb{Z}_{n_1}$ and $\mathbb{Z}_{n_2}$, where $n_1 \cdot n_2 = \#E_p(a, b)$ and n_2 divides n_1 and also divides $p - 1$. The first lemma to find the order of a special elliptic curve is that for an odd prime p satisfying $p = 2 \pmod 3$ and b satisfying $0 < b < p$, the order of $E_p(0, b)$ is

$$\#E_p(0, b) = p + 1 \tag{12}$$

The second lemma is that for a prime p satisfying $p = 3 \pmod 4$ and a satisfying $0 < a < p$, the order of $E_p(a, 0)$ is

$$\#E_p(a, 0) = p + 1 \tag{13}$$

3.2. Multiplication

Multiplying a point P on an elliptic curve by a scalar k is defined as adding P to itself k times. This can be computed efficiently with an algorithm similar to the algorithm for computing large powers. One first convert k to base-2. Then, one repeatedly adds P to itself using the formula described in the previous section to get $2i \cdot P$. One adds this to the sum if the ith digit of the base 2 representation of k is 1. One will only need to repeat this process $\log(t)$ times to get to $k \cdot P$, and the complexity of this algorithm is $O(\log t)$ which is significantly faster than the original $O(k)$ time that it would have taken to add P to the sum k times. Since there is no operation to simply multiply a point by a scalar in $O(1)$ time, the complexity of this is equivalent to exponentiation over a finite field. Like the discrete logarithm problem, this operation is also used to ensure the security of cryptosystems because it is possible to efficiently calculate $R = k \cdot P$ but extremely hard to find k from only knowing R and P. The best-known algorithms to solve the discrete log problem on elliptic curves have exponential run times, and this better supports the reliability of the trapdoor function on which the article is based.

3.3. Elliptic curve Diffie-Hellman

The elliptic curve Diffie Hellman is like the regular Diffie Hellman in that the sender, Alice, wants to communicate with Bob over an insecure channel, and they use the elliptic curve Diffie Hellman over a prime field F_p.

First, Alice and Bob agree on a set of parameters (p, a, b, P, n, h), where p is a prime, a, b are the numbers that make up the coefficients of the equation of the elliptic curve E, P is a point on E, n is the order of the cyclic subgroup generated by P, and h is a cofactor of the group G on E from which P is chosen. Now, the cyclic subgroup generated by the point P consists of the point at infinity O all the points that can be written as kP, where k is a constant that satisfies $1 \leq k \leq n-1$. n should be a large number to make the discrete logarithm harder to solve, h should be a small number, preferably 1. Often some organizations use curves with pre-computed parameters so that the sender and recipient do not have to calculate them since this can be quite time-consuming. These parameters are public and known by everyone.

Now, Alice chooses an integer d_a that satisfies $1 \leq d_a \leq n-1$ and uses it as her private key. She also calculates a point $Q_a = d_a P$ which is her public key. Then Bob also chooses a private key d_b and calculates a public key $Q_b = d_b P$. Then Bob and Alice exchange their public keys. Note that because of the discrete logarithm problem, others are unable to calculate the private keys from the public keys.

Now Bob multiplies Alice's public key by his private key to get $S = d_b Q_a$ and Alice multiplies Bob's public key by her private key to get $S = d_a Q_b$. These two expressions are equal since

$$d_b Q_a = d_b d_a P = d_a Q_b = S \tag{14}$$

S is now the shared key, and a third party is unable to calculate this key because of the discrete logarithm problem. Alice and Bob can now use this key to efficiently communicate over the insecure channel.

3.4. El Gamal System on elliptic curves

For the El Gamal System, the parameters are p, a, b, P, n, where p is a prime, a, b are the coefficients of an elliptic curve E that one choose over the field F_p, P is a point on E, and n is the order of the cyclic subgroup generated by P. Bob now chooses a number d as his private key and multiplies the point P by d to calculate a public key $Q = dP$ like in the Diffie Hellman key exchange.

First, Alice maps her message m to a point M on the elliptic curve E using a function $f(m)$. Then she chooses a number k from 1 to $n-1$ and calculates the point $C = kP$. Finally, she computes another point $D = M + kQ$ Now, Alice sends the pair (C, D) to Bob as her ciphertext.

Now, Bob can decrypt the ciphertext by first computing $M = D - dC$. This works because $C = kP$ and $Q = dP$, so

$$dC = dkP = kQ \tag{15}$$

and the expressions cancel out when subtracted. Now, Bob can simply compute $f^{-1}(M)$ to get to the message m.

Note that this system is secure because in order for a third party to get to M from knowing only D and kP, they would need to find $kQ = dkP$ which is the discrete logarithm problem.

4. Cryptography using elliptic curves over $\mathbb{Z}_n$

4.1. Elliptic curves over a ring

Now one can consider an elliptic curve over a ring, $\mathbb{Z}_n$, where n is an odd composite squarefree integer which means that none of the prime factors of n have an even degree [1, 3]. This curve can be defined as the set of all pairs (points) $(x, y) \in \mathbb{Z}_n^2$ that satisfy $y^2 = x^3 + ax + b$ and also the point at infinity O, and is represented here as $E_n(a, b)$. Addition of two elements of $E_n(a, b)$ is defined the same as over a field F_p, where the same computations are performed over $\mathbb{Z}_n$ instead of over F_p.

However, there are two problems with this. The first is that division over $\mathbb{Z}_n$ is not always defined. This is because division by an integer r mod n is defined as multiplication by x, the inverse of r which is another integer that satisfies $xr = 1 \bmod n$. As one can see, this inverse x only exists when r and n are relatively prime to each other, and since n is no longer a prime, there will be some numbers in the set $\mathbb{Z}_n$ that do not have an inverse. Addition is also not always defined because as one can see from the previous section, calculations for the values of (x, y) of the third point involves a value λ that is only defined when the divisor is nonzero. The second problem is that the set $E_n(a, b)$ is not a group. However, there are solutions to these problems.

Suppose that n is the product of two primes p and q. One can calculate operations of an elliptic curve mod n by performing the operations of the elliptic curve mod p and mod q separately. By the Chinese remainder theorem, one can then represent each element P on $E_n(a, b)$ as the pair (P_p, P_q), where P_p is a point from the set $E_p(a, b)$ and P_q is a point from the set $E_q(a, b)$. Now all the points in the set $E_n(a, b)$ can be represented except for those where exactly one of the two points P_p and P_q is the point at infinity. Addition of two elements on $E_n(a, b)$ is also not defined when the resulting point is one of these points. However, when the prime factors of n are very large, the probability of this actually happening is extremely small since there are lots of points total. Therefore, this problem can be disregarded because of its unlikeliness.

One can solve the second problem (that $E_n(a, b)$ is not a group) by using the Chinese remainder theorem and finding a group on the elliptic curve. One can choose a point P on $E_n(a, b)$ and repeatedly add it to itself until the result comes back to P. Then one can use all the points that can be represented by adding P to itself a certain number of times, and the resulting set is a cyclic group. Now one can use the properties of groups on this new cyclic group and disregard the other points on $E_n(a, b)$ that are not in the group. Although it is possible to define an elliptic curve over a ring so that it is always a group, this would involve adding in the terms cy and dx^2 in the original definition of the elliptic curve, and is unnecessary for our purposes.

4.2. Cryptography

Suppose user A wants to send user B a secret message. In order to generate a key, user A first chooses large primes p and q that satisfy $p \equiv q \equiv 2 \bmod 3$ and uses these to compute $n = pq$, and $N_n = lcm\left(\#E_p(0, b), \#E_q(0, b)\right)$, which is equal to $lcm(p + 1, q + 1)$ according to the lemma from earlier. Then, A chooses an integer e coprime to N_n and finds an integer d that satisfies

$$ed \equiv 1\ (mod\ N\_n) \tag{16}$$

A's secret keys are now d, p, q, N_n, $\#E_p(0, b)$, and $\#E_q(0, b)$, and A's public keys are n and e.

Then, to encrypt a plaintext $M = (m_x, m_y)$ that satsfies $m_x, m_y \in \mathbb{Z}_n$, one assume that the point M is a point on the elliptic curve $E_n(0, b)$, where b is a number determined by m_x and m_y. Now, user A encypts the point M using the public keys e and n with the function

$$C = e \cdot M \text{ over } E_n(0, b) \tag{17}$$

The result of this is a ciphertext pair $C = \left(c_x, c_y\right)$, and user A sends C to user B.

User B can now decrypt the message using the public key n and his secret key d by using the function

$$M = d \cdot C \text{ over } E_n(0, b) \tag{18}$$

5. Quantum attacks

5.1. Shor's algorithm

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations [6, 7].

Theorem 1. Suppose x is a non-trivial solution to the equation $x^2 \equiv 1 \bmod N$ where N is an L bits composite number and $1 \leq x \leq N$. One can apply $O(L^3)$ operations to find one of $\gcd(x-1, N)$ and $\gcd(x+1, N)$ as a non-trivial factor of N.

Lemma 1. Let p be an odd prime. Let 2^d be the largest power of 2 dividing $\phi(p^\alpha)$. Then with probability exactly one-half 2^d divides the order modulo p^α of a randomly chosen element of $\mathbb{Z}_{p^\alpha}^*$.

Theorem 2. Suppose N is an odd composite positive integer and y is a number co-prime to N, it is likely that $y^{r/2} \neq \pm 1 \bmod N$, and $x \equiv y^{r/2} \bmod N$ is a nontrivial solution to $x^2 \equiv 1 \bmod N$,

$$\Pr\left\{r \text{ is even and } y^{r/2} \neq \pm 1 \bmod N\right\} \geq 1 - \frac{1}{2^m} \tag{19}$$

where is m is the number of prime factors of N and r is the order of y.

Theorem 1 and Theorem 2 combined develops an algorithm of reduction of factoring to order-finding.

a) If N is even, return the factor 2.

b) If $N = a^n$ for $a \geq 1$ and $b \geq 2$, return the factor a.

c) For a randomly chosen number x such that $1 \leq x \leq N-1$, return $\gcd(x, N)$ if it is greater than 1.

d) Find the order r of x modulo N using order-finding

e) If r is even and $x^{r/2} \neq \pm 1 \bmod N$ then compute $\gcd\left(x^{r/2} - 1, N\right)$ and $\gcd\left(x^{r/2} + 1, N\right)$. Test and return if one of these is a non-trivial factor. If not, the algorithm fails.

The goal of Shor's algorithm is to factor any large number within a reasonable amount of time. With this, one is able to find the primes p and q the N consists of in the elliptic curve algorithm and the private key d as well. Therefore, the elliptic curve algorithm can be broken by a quantum computer in time $O(\log n)$ and becomes no longer secure.

6. Conclusion and discussion

The findings in this article are expected to contribute to the broader understanding of potential quantum-resistant cryptographic systems and to help in developing better security measures to safeguard sensitive information in the post-quantum era. The exploration of elliptic curve cryptography over non-prime rings represents a first crucial step towards future-proofing our sensitive information against the incoming challenge posed by quantum computing. Although quantum computers are not developed enough to run Shor's algorithm yet, they will be in the near future, and using Shor's algorithm, it is possible to reverse the one-way functions that the ellpitic curves over non-prime rings rely on for their security. When quantum computers are fully developed, this method will be vulnerable to quantum attacks and therefore no longer secure. Therefore, other broader and newer ideas are essential.

This article is not without its limitations. Firstly, this paper is based on Shor's algorithm in quantum computing. As quantum information is a rapidly developing field, there might have been newer algorithms or hardware beyond the current knowledge. Secondly, although the analysis of attacks on elliptic curves over non-prime ring is rigorous, it is only a theoretical investigation. A fully tested and carefully simulated computer program is beyond the scope of the article. Last but not least, all practical concerns such as costs and accessibility of quantum computing have been ignored. The attackers has been given the benefit of having the full power of quantum computing as their fingertips. This is to ensure the utmost security standard in our study.

In order to future-proof sensitive information, further exploration has to be made until we achieve post-quantum cryptography. Post-quantum cryptography refers to cryptographic systems resistant to quantum computers but are themselves operating on classical computers. In order to find such systems, one must explore new mathematical structure and hardness assumptions able to withstand quantum attacks.

References

[1] Sala, Massimiliano, and Daniele Taufer. ”The group structure of elliptic curves over Z/NZ.” arXiv:2010.15543 (2020).

[2] Pradella, S. ”Introduction to Elliptic Curve Cryptography.” (2000).

[3] Koyama, Kenji, et al. ”New public-key schemes based on elliptic curves over the ring Z n.” Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1991.

[4] Silverman, J. H., and Tate, J. T. (1992). Rational points on elliptic curves (Vol. 9). New York: Springer-Verlag.

[5] Blake, I., Seroussi, G., Seroussi, G., & Smart, N. (1999). Elliptic curves in cryptography (Vol. 265). Cambridge university press.

[6] Nielsen, Michael A., and Isaac Chuang. ”Quantum computation and quantum information.” (2002): 558-559.

[7] Preskill, John. ”Lecture notes for Physics 219: Quantum computation.” Caltech Lecture Notes (1999).