Computational Problems For Designing Asymmetric Cryptosystems

Ganyuan Cao[1]

Instructor: Andrew Bremner

Arizona State University, Tempe, AZ, 85281, USA
Ganyuan.Cao@asu.edu

Abstract. Cryptography is a critical study which impacts a variety of fields from keeping top secrets to everyday checking-account transitions. Basically, cryptography is about encryption and decryption. To meet the development of modern society, cryptography has evolved and developed its four main objectives: confidentiality, data integrity, authentication, and non-repudiation. This paper briefly described the development of asymmetric cryptosystems in modern cryptography. This paper focuses on the discussion regarding computational problems which are used to design modern asymmetric cryptosystems. The paper analyzed the security of current cryptosystems under the attack of modern computers and briefly discussed the security of cryptosystems under the potential attacks of quantum computers by analyzing the algorithms to solve those computational problems.

Keywords: Asymmetric Cryptography · Computational Problems · Abstract Algebra · Post-Quantum Cryptography

1 Introduction

Cryptography, the study of encryption and decryption, has a long history of application for secure communication. Cryptography is mainly divided into classical ciphers and modern cryptosystems. Classical ciphers are designed mainly based on the substitution of letters in the alphabet. As computer and network technology develop, classical ciphers are not used anymore for the reason of insufficient security under the attack of modern computers. To meet the need for higher security, modern cryptosystems are designed based on the difficulty of solving the computational problems involving large prime numbers, which is proved to resist the attack with modern computers. The application of modern cryptography has become wider, not only for the military but also for civil use. Cryptography, no matter classic ciphers or modern cryptosystems, has a strong connection with mathematics including number theory, abstract algebra, probability, and statistics. This paper will evaluate some modern asymmetric cryptosystems based on their mathematical background. We define the notations $E_{key}(message)$ for encrypting message using key and $D_{key}(message)$ for

decrypting message using *key*. The organization of the paper is below: In the second section, the development of asymmetric cryptography will be discussed. In the third section, asymmetric cryptography and its application will also be discussed. In the fourth section, computational problems applied in modern asymmetric cryptography will be evaluated and discussed. In the fifth section, we made the conclusion on computational problems which are used to design asymmetric cryptosystem from the complexity view in computational theory and discussed potential mathematical theories which may be useful for designing new asymmetric cryptosystem.

2 Development of Asymmetric Cryptography

Cryptography, as an art, has been existing for long. Many famous classical ciphers were designed for transmitting information during a war, for example, Ceaser Cipher, Hill Cipher, and the famous Enigma in World War II. As computer and communication technology develops, classical ciphers can be broken easily and the need for cryptography in daily use was quickly increasing. With such a background, the concept of modern cryptography was proposed. In modern cryptography, there are two aspects: Asymmetric cryptosystems (also called public-key cryptosystems) and symmetric cryptosystems.

Asymmetric cryptosystems are designed for the purpose of establishing secure communication without a secret key which is known by both parties before the communication. Asymmetric cryptosystems play an important role in establishing secret keys which are used in symmetric cryptosystems i.e., key exchange. Public key cryptosystems are mainly designed using the difficulty of computational problems involving large prime numbers, for example, RSA and ElGamal cryptosystem.

3 Asymmetric Cryptography and Its Application

Asymmetric Cryptography makes it possible to establish secure communication without a pre-shared secret key, which satisfies the need caused by the development of the Internet. In Asymmetric Cryptography, a key pair of public key and secret key is generated, say (P_A, S_A) for "Alice",(P_B, S_B) for "Bob" . The public key is made public to all participants while the secret key is just kept by the user who generated the key pair only.

Encryption with Asymmetric Cryptography is by public key and decryption is done through the user's secret key. In this way, asymmetric cryptography provides secure communication without a pre-shared secret key. For example, Bob wants to send Alice an encrypted message m. Bob encrypts the message with Alice's public key to obtain the ciphertext $c = E_{P_A}(m)$. Then Alice decrypts the ciphertext with her secret key to get the plaintext $m = D_{S_A}(c)$.

Also, Asymmetric Cryptography provides the function to sign a digital signature which makes it possible to verify if a certain message was sent by a certain party. In this way, user's non-repudiation was guaranteed by the Asymmetric

Cryptography. The digital signatures are usually made by the sender's secret key while the verification process is done using the sender's public key. For example, Alice wants to sign a document m. She made $Sig = E_{S_A}(m)$. Bob can verify the signature by $m = D_{P_A}(Sig)$. If Bob can decrypt the signature Sig with Alice's public key, then it shows that the signature is from Alice.

Asymmetric Cryptography is often used to transmit file(message) of relatively small size for the reason that certain messages must be converted (or partitioned) to large numbers to be encrypted or decrypted. Thus Asymmetric Cryptography is usually used to transmit the key which is used in symmetric cryptography such as AES to encrypt the file with big size as symmetric cryptography encrypts the file bitwise, which is more efficient in a computer. In this section, we will discuss three applications of Public Key Cryptography including Encryption, Digital Signature and Key Infrastructure with some famous public key cryptography algorithms.

3.1 RSA Cryptosystem

RSA cryptosystem [1] is one of the first public key cryptosystem. The design of RSA algorithm is based on the difficulty of factorization of a composite number which is a product of two large primes. The public key pair in RSA algorithm is (n, e) where n is the product of two large primes and e is the encryption exponent.

Algorithm 1 RSA Algorithm

1: Alice chooses prime p, q, compute $n = pq$ and publish n.
2: Alice computes encryption exponent $e : gcd((p-1)(q-1), e) = 1$ and publish e.
3: Compute decryption exponent $d : de \equiv 1 \pmod{(p-1)(q-1)}$ and keep d secret.
4: Bob encrypts plaintext m to ciphertext $c : c \equiv m^e \pmod{n}$ and sends c to Alice
5: Alice decrypts ciphertext c to plaintext $m : m \equiv c^d \pmod{n}$.

RSA is also used in digital signature which is used to verify the identity of sender and guarantee non-reputation of sender. Instead of using public key to encrypt, RSA signature uses secret key to sign a certain message. However, the secret key of sender can not be derived from the signature as it is a discrete logarithm problem to derive the sender's secret key from the plaintext and signatures which is of great difficulty.

Algorithm 2 RSA Signature

1: Alice chooses prime p, q, compute $n = pq$ and publish n.
2: Alice computes public key $e : gcd((p-1)(q-1), e) = 1$.
3: Alice computes secret key $d : de \equiv 1 \pmod{(p-1)(q-1)}$.
4: Alice signs plaintext m to signature $s : s \equiv m^d \pmod{n}$ and sends s to Bob.
5: Bob verifies the signature $s : m \equiv s^e \pmod{n}$.

Theorem 1 (Euler's Theorem). *If n and a are relatively positive primes, then* $a^{\phi(n)} \equiv 1 \pmod{n}$

We use Euler's Theorem to justify the correctness of RSA. By RSA algorithm,

$$c^d \equiv (m^e)^d \equiv m \pmod{n}$$

The design of RSA is based on the factorization of composite which is product of two large primes. If p, q are obtained, then $\phi(n)$ can be obtained,which results the acquirement of the secret key d.

3.2 Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange is a key infrastructure protocol based on discrete logarithm problem. Diffie-Hellman Key Exchange is widely used in exchanging the secret key used in symmetric cryptosystems. Diffie-Hellman Key Exchange establishes the condition for secure communication without two parties meeting in person to exchange secret key.

Algorithm 3 Diffie-Hellman Key Exchange

1: Alice and Bob choose a large prime p and a primitive root a modulo p.
2: Alice chooses a secret integer i, send $a^i \pmod{p}$ to Bob.
3: Bob chooses a secret integer j, send $a^j \pmod{p}$ to Alice.
4: Alice compute $y \equiv (a^i)^j \pmod{p}$ to obtain the shared message.
5: Bob compute $y \equiv (a^j)^i \pmod{p}$ to obtain the shared message.

3.3 ElGamal Cryptosystem

ElGamal cryptosystem [2] is based on the difficulty of solving a discrete logarithm problem in a multiplicative group, say $\mathbb{Z}_p^\times$. The public key pair in ElGamal cryptosystem is (p, α, β) where p is a large prime and α is a primitive root modulo p.

Algorithm 4 ElGamal Cryptosystem

1: Choose a large prime p, and a primitive root α modulo p.
2: Choose a secret integer a such that $a \not\equiv 1 \pmod{p-1}$.
3: Compute $\beta \equiv \alpha^a \pmod{p}$.
4: The public key is the tuple (p, α, β).
5: Choose random integer k and compute $r \equiv \alpha^k \pmod{p}$.
6: With plaintext m , compute $t \equiv \beta^k m \pmod{p}$.
7: The encrypted text is (r, t).
8: With ciphertext (r, t), compute $m \equiv tr^{-a} \pmod{p}$ to obtain plaintext m.

3.4 ECC : Elliptic Curves Cryptography

Elliptic Curves Cryptography is mainly based on on discrete logarithm problem with additions of points on elliptic curves. In general, Elliptic Curves Cryptography can implement the cryptography algorithms based on discrete logarithm problem such as ElGamal Cryptosystem and Diffie-Hellman Key Exchange by converting the discrete logarithm problem in multiplicative group into scalar multiplication(repeatedly addition of points) of points on elliptic curves. Here we want to discuss the Massey-Omura Cryptosystem[4] which is designed based on elliptic curves. Massey-Omura is a three-pass protocol which means that two parties must complete three handshakes in order to complete the message exchange.

Algorithm 5 Massey-Omura Cryptosystem

1: Alice and Bob agree on an elliptic curve E over finite field $\mathbb{F}_q$. Let $N = |E(\mathbb{F}_q)|$, $P = (x, y)$ which represents the message m.
2: Alice and Bob each secretly pick random a and b such that $gcd(a, N) = 1$ and $gcd(b, N) = 1$.
3: Alice computes $[a] \cdot P$, and sends to Bob.
4: Bob computes $[b] \cdot ([a] \cdot P)$, and sends to Alice.
5: Alice and Bob compute $a^{-1}, b^{-1} \in \mathbb{Z}^+$.
6: Alice computes $[a^{-1}] \cdot ([b] \cdot ([a] \cdot P))$, and sends to Bob.
7: Bob compute $[b^{-1}] \cdot ([a^{-1}] \cdot ([b] \cdot ([a] \cdot P))) = m$

4 Computational Problems for Modern Cryptography

In this section, we mainly discussed the computational problems which cryptosystems discussed above are based on to preserve security under certain attacks and the algebra structures the computational problems involve. In this section, we also discussed the Lattice-based Cryptography which is a candidate for post-quantum cryptography and several computational problems related to lattices.

4.1 Factorization of composite numbers

RSA is based on two mathematical problems : factorization of large composite numbers and a modular exponent problem. The most efficient way to break RSA is to factorize the public key of RSA, say $n = pq$ be a product of two large primes p, q. Consider that by the key generation algorithm of RSA, the decryption key is generated by $de \equiv 1 \pmod{\phi(n)}$ i.e., d is a multiplicative inverse of e modulo $\phi(n)$. With the decryption key d, the attacker could retrieve the plaintext directly by computing $c^d \equiv m \pmod{n}$. Thus factorization of integer became the most efficient way to break the RSA cryptosystem.

To be against the attack of different factorization algorithms, the choices of prime numbers to form RSA public key have been discussed. We take *Pollard's P-1 Factorization*[9] as example. In generating RSA public key, $p-1$ and $q-1$ are required to contain a factor which is large prime number. The primes p, q are obtained by choosing two pair numbers p_0, q_0, computing $p = kp_0 + 1$,$q = kq_0 + 1$ for $k \in \mathbb{Z}^+$ and being tested for primality. Consider that *Pollard's P-1 Factorization* can succeed with the choice of bounder β where $p_0 < \beta < q_0$. In current RSA cryptosystem, a key of 2048 bits is recommended by NIST. With relatively large prime p_0, q_0, it is infeasible for current computers to factorize the number $n = pq$ in effective time.

As the computing technology and quantum computation develop, a 2048-bit key may not be secure for RSA after 2030. According to NIST, a 3072-bit key is recommended for RSA after 2030. Here we also want to discuss Shor's Algorithm[5] with quantum computation in prime factorization. Shor's Algorithm has time complexity $O((\log n)^2(\log\log n)(\log\log\log n))$, which makes prime factorization feasible with quantum computation.

4.2 Discrete Logarithm Problem

In $\mathbb{R}$, a logarithm problem is to find the number x such that $b^x = a$, then $x = \log_b a$. In discrete logarithm problem, we define a group G under multiplicative operation with identity e_G, define the discrete logarithm by find element $x \in \mathbb{Z}^+$ such that for $a, b \in G$, $b^x = a$. In Number Theory, Index Arithmetic is more commonly used than discrete logarithm problem. Say that $m \in \mathbb{Z}$ and r a primitive root modulo m, then $x = ind_r(a)$ if $r^x \equiv a \pmod{m}$ and $gcd(a, m) = 1$. In designing cryptosystem, the exponent of discrete logarithm is usually used as the secret key for the property that x is hard to find with only (a, b).

In cryptography, the multiplicative cyclic groups $\mathbb{Z}_p^\times$ where p is prime are usually used. Also, a primitive root modulo p is often used to ensure the worst-case for attackers. We use ElGamal Cryptosystem to analyze the discrete logarithm problem used in cryptography. Consider that Alice and Bob agree on a primitive root α modulo p where p is a large prime. There are two ways for an attacker to retrieve the plaintext. The first way is to obtain the secret key of Alice by computing $\alpha^a \equiv \beta \pmod{p}$. The second way is to find the k such that $r \equiv \alpha^k \pmod{p}$. In two scenarios, the attacker would have to solve a discrete logarithm problem which is computational infeasible with relatively large prime.

The fastest algorithm for finding discrete logarithm modulo prime is proposed by Gordon [6] in 1993 using number field sieve.The algorithm runs in $exp(O(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}})$.With industrial primes, cryptosystems based on discrete logarithm problem are computationally infeasible.

Although ElGamal Cryptosystem and other cryptosystems based on discrete logarithm problem over $\mathbb{Z}_p^\times$ are computationally infeasible with current computers, those cryptosystems may also be broken as the quantum computers develop. Peter W. Shor proposed the algorithm [5] to find discrete logarithms with two modular exponentiations and two quantum Fourier transforms. The algorithm proposed by Shor can be done in polynomial if the attacker try all proper candidates,

which means the discrete logarithm problem which could not be solved by current computers may be solved by algorithms implemented on quantum computers in the future.

4.3 Addition of points on Elliptic Curves

Elliptic Curves over finite field are finite abelian groups. We use short Weierstrass Equation to denote an elliptic curve: $E : y^2 = x^3 + Ax + B$ where A, B are coefficient. Fix two points $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ on the elliptic curve E. We define the group law of addition of elliptic curves below.

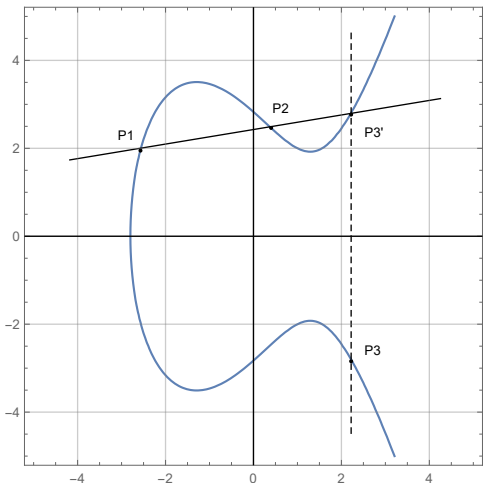

Fig. 1. Addition of points on elliptic curves

1. Draw a line L through point P_1, P_2 intersects the elliptic curve at P_3'.
2. Reflect P_3' across x-axis to obtain $P_3 = (x_3, y_3)$.
3. Define $P_3 = P_1 + P_2$

Elliptic Curve is the algebraic structure that can implement multiple asymmetric cryptosystem such as ElGamal Cryptosystem and Diffie-Hellman Key Exchange. Also, cryptosystems over elliptic curves are mostly based on the discerte logarithm problem, which is the multiplication of points on elliptic curves.

Elliptic curves over finite fields $E(\mathbb{F}_q)$ are abelian groups with additive operation[8]. With $a \in \mathbb{Z}^+$, the multiplication of point P with a, denoted, $[a] \cdot P$ is considered as a discrete logarithm problem over the elliptic curves. We use Massey-Omura Cryptosystem to analyze the difficulty of discrete logarithm

problem with elliptic curves. In Massey-Omura Cryptosystem, two parties establish a public elliptic curve over a finite field, say $E(\mathbb{F}_q)$, and the order of the elliptic curve $N = |E(\mathbb{F}_q)|$ is also public. The secret keys in Massey-Omura Cryptosystem are a, b for Alice and Bob, which satisfies the condition that $gcd(a, N) = gcd(b, N) = 1$.The encryption process for Massey-Omura Cryptosystem is done by computing the multiplication of point (which represents the message) with secret keys of Alice and Bob i.e., $[a] \cdot P$ and $[b] \cdot P$. The decryption process for Massey-Omura Cryptosystem i done by computing the multiplication of point with the inverse of a and b, say $a^{-1}, b^{-1} \in \mathbb{Z}_N^{\times}$.With known secret key, Alice and Bob can encrypt and decrypt the message efficiently. Also, Massey-Omura Cryptosystem is considered as a three-pass protocol,which meas that the exchange of message requires three handshakes. Suppose that an attacker intercepted a message encrypted with Massey-Omura Cryptosystem. Without loss of generality, the attacker may have intercepted $[a] \cdot P$, $[a] \cdot ([b] \cdot P)$ or $[a^{-1}] \cdot ([a] \cdot ([b] \cdot P))$. In each scenario, the attacker would have to solve a discrete logarithm problem to retrieve the plaintext. Current ways for solving discrete logarithm problem of elliptic curves including non-generic method such as index calculus and generic method such as *Baby-Step-Giant-Step*. Consider that $N = |E(\mathbb{F}_q)|$, by Hasse's Theorem, we have that the N falls in the range

$$q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}$$

Hasse's Theorem leads to the conclusion that *Baby-Step-Giant-Step* to solve discrete logarithm problem of elliptic curves is $\mathscr{O}(\sqrt[4]{q})$. The choice of finite field is critical in Elliptic Curves Cryptography. For example, SECP256K1 uses $q = 2^{256} - 2^{32} - 977$ and NIST P-224 uses $q = 2^{224} - 2^{96} + 1$. It can be concluded that discrete logarithm of elliptic curves over finite fields of industrial primes above is computationally infeasible with current computers.

4.4 Computational Problem with Lattice

Lattice is an abstract structure in order theory. Lattice-based cryptography[10] is considered as a candidate for post-quantum cryptography which means that lattice-based cryptography provides certain security under the great computational power possessed by quantum computers. Define a lattice $\mathscr{L} \subset \mathbb{R}^n$:

$$\mathscr{L} = \mathscr{L}(\mathbb{B}) = \{\sum_{i=1}^{n} a_i \boldsymbol{V_i} : a_i \in \mathbb{Z}\}$$

where $\mathbb{B}$ is the basis of lattice $\mathscr{L}$ and $\mathbb{B} = (\boldsymbol{V_1}, \boldsymbol{V_2}, \cdots, \boldsymbol{V_n})$.

Current lattice-based cryptography is designed from the difficulty of solving computational lattice problems including *Shortest Vector Problem*(SVP), which means that the output of the problem is the shortest nonzero vector in a lattice. The fact that the given lattice basis contains much longer vectors than the shortest nonzero vector makes SVP a computational problem, specifically NP-Hard problem. Lattice-based cryptography includes hash function, public

key encryption scheme and digital signature scheme. We specifically discuss GGH/HNF public key cryptosystem[11] here. In GGH/HNF cryptosystem, a "good" lattice basis $\mathbb{B}$ (a basis consisting of short, almost orthogonal vectors) is used as the secret key. A "bad" lattice basis $\mathbb{H}$ such that $\mathscr{L}(\mathbb{H}) = \mathscr{L}(\mathbb{B})$ is used as the public key. Then encryption process is processed by adding a short noise vector $\boldsymbol{r}$ to a chosen lattice point $\boldsymbol{v}$, say the resulting ciphertext is $\boldsymbol{c} = (\boldsymbol{r} + \boldsymbol{v})$. The decryption is processed by finding the lattice point $\boldsymbol{v}$ which is closest to $\boldsymbol{c} = (\boldsymbol{r} + \boldsymbol{v})$ to obtain the plaintext $\boldsymbol{r} = \boldsymbol{c} - \boldsymbol{v}$. Consider that in GGH/HNF cryptosystem, a "good" lattice basis is kept secret for the receiver, which allows the receiver to compute the vector $\boldsymbol{v}$ efficiently while a "bad" lattice basis is kept public which does not affect the security of the secret basis. In the view of cryptoanalysis, the "bad" basis is usually considered the worst possible basis for $\mathscr{L}(\mathbb{B})$ which means that it is computationally infeasible to obtain $\mathscr{L}(\mathbb{B})$ with the basis $\mathbb{H}$.

Current lattice-based cryptography is divided into two part. The first part is based on worst-case hardness problems associated with lattice, however, those cryptosystems usually lack efficiency which becomes impractical in real life. The second part is practical and efficient but the security of those cryptosystems should still be evaluated to find proof to support their security. However,lattice-based cryptography is still considered as a good candidate for post-quantum cryptography because there are currently no efficient quantum algorithm to solve lattice problems.

5 Conclusion

P verus NP problem is a major problem in mathematics and computer science. P problems are those problems that can be solved in polynomial time while NP problems are those problems that can be verified in polynomial time. Current asymmetric cryptosystems are designed based on NP problems. Encryption is a P problem while decryption is the NP problem. With public key, one can encrypt a message easily. On the other hand, with the solution(secret key), one can verify(decrypt the message) easily but one can not solve the problem(break the ciphertext) directly.

In the paper, we included the NP problems which are used in designing current asymmetric cryptosystem including prime factorization, discrete logarithm problem and addition of points on elliptic curves. There are also many other asymmetric cryptosystems which are designed based on NP problem in other mathematical theories. For example, Knapsack-type cryptosystem[7], which is considered as a good candidate for post-quantum cryptography, is designed based on 0-1 Knapsack problem which is in the theory of combinational optimization. As the quantum computation develops, current asymmetric cryptosystems should be evaluated of security under the attack of quantum algorithms. Current cryptosystems may need modification such as padding or modifying the key generating process to increase the difficulty to be secure under the attack of quantum computation. New asymmetric cryptosystems should also be explored

from current NP problems in different theories of mathematics and computer science to ensure the security of communication in post-quantum cryptography.

References

1. Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21.2 (1978): 120-126.
2. ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." IEEE transactions on information theory 31.4 (1985): 469-472.
3. Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." IEEE transactions on Information Theory 22.6 (1976): 644-654.
4. Massey, James L., and Jimmy K. Omura. "Method and apparatus for maintaining the privacy of digital messages conveyed by public transmission." U.S. Patent No. 4,567,600. 28 Jan. 1986.
5. Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM review 41.2 (1999): 303-332.
6. Gordon, Daniel M. "Discrete Logarithms in GF(P) Using the Number Field Sieve." SIAM Journal on Discrete Mathematics 6.1 (1993): 124-138.
7. Chor, Benny, and Ronald L. Rivest. "A knapsack-type public key cryptosystem based on arithmetic in finite fields." IEEE Transactions on Information Theory 34.5 (1988): 901-909.
8. Washington, Lawrence C. Elliptic curves: number theory and cryptography. Chapman and Hall/CRC, 2003.
9. Pollard, John M. "Theorems on factorization and primality testing." Mathematical Proceedings of the Cambridge Philosophical Society. Vol. 76. No. 3. Cambridge University Press, 1974.
10. Micciancio, Daniele. "Lattice-based cryptography." Encyclopedia of Cryptography and Security (2011): 713-715.
11. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In Advances in cryptology, volume 1294 of Lecture Notes in Comput. Sci., pages 112–131. Springer, 1997.