A comparative analysis of public key cryptographic algorithms: RSA, ELGamal, and elliptic curve encryption

Yafei Xie

Department of Computer Science, University College London, London, WC1E 6BT, UK

oliver.xie.22@ucl.ac.uk

Abstract. Cryptography stands as an indispensable and efficient facet within the expansive field of information security. It offers a reliable method for ensuring the confidentiality and integrity of data during the complex process of information transmission between a sender and a recipient. Beyond this, exclusive decryption privileges are meticulously reserved for the designated recipient. This individual holds the exclusive authority to decipher the transmitted information, which has been encrypted by the key holder beforehand. This scholarly investigation introduces and delves into three prevalent cryptographic algorithms: RSA, El-Gamal, and Elliptic Curve Cryptography. It offers a discerning examination and contrast of the underlying mathematical challenges associated with each sophisticated method. The discourse unfolds a detailed comparative analysis of these three pivotal algorithms, zeroing in on their crucial aspects such as key size length and operational running time. The in-depth exploration within this study aims to shed light on the intricate workings, strengths, and potential limitations of RSA, El-Gamal, and Elliptic Curve Cryptography. By unraveling these aspects, the study contributes to a richer understanding and more informed choices in the practical application of cryptographic algorithms, enhancing the overarching realm of information security in an increasingly digital and interconnected world.

Keywords: RSA, ElGamal, ECC.

1. Introduction

Cryptography stands as an essential pillar within the domain of information security. It embodies the practice of intricately encoding and decoding data to ensure exclusive comprehension and manipulation by authorized entities during both storage and transmission. The strategic deployment of cryptographic protocols substantially diminishes the likelihood of unauthorized access to confidential communications, thereby reinforcing the privacy of users [1].

The escalating ubiquity of the Internet and digital devices underscores the paramount importance of robust information security. Cryptography emerges as a critical security mechanism, meticulously safeguarding message transmission, document storage, digital signatures, and certificates [2]. Its diverse applications steadfastly aim to bestow confidentiality, integrity, availability, and non-repudiation upon the information, bolstering its security across various platforms.

The realm of cryptography bifurcates into two distinct categories: Private (symmetric) and Public (asymmetric) key cryptography. In a symmetric key cryptosystem, two parties employ an identical shared key for both encryption and decryption processes. Conversely, in a public key cryptosystem, each party possesses individual public and private keys, referred to as "p" and "s" respectively. The key "s," known solely by the party itself, governs the decryption process, while the publicly shared key "p" oversees encryption.

In a typical scenario utilizing asymmetric cryptography, if party A desires to confidentially share plaintext with party B, A employs B's public key alongside the plaintext to generate a ciphertext. Upon receipt, B employs their private key to decipher the message, ensuring secure communication without a shared secret key, as innovatively proposed by Diffie and Hellman in 1976 [3]. Public key cryptosystems, including widely utilized algorithms like RSA, ElGamal, and ECC, obviate the necessity for a secure channel or advance secret key sharing, offering a significant advantage over private key systems and diminishing the logistical and financial burdens of secret key distribution and management. This paper embarks on a comparative exploration of these three prominent public key algorithms, evaluating diverse aspects to ascertain their performance. It encompasses an overview of the three public key cryptosystems and introduces and compares the fundamental computational challenges for each algorithm, culminating in a comprehensive comparative analysis, centered on key size length and operational time.

2. Algorithmic Details

2.1. RSA

In 1977, Rivest et al. presented the idea of RSA algorithm [4]. The RSA algorithm is an example of an asymmetric cryptographic technique that is often implemented for message encryption, decryption and digital signature. The provision of a digital signature via the use of the public key in the RSA encryption algorithm confers upon it a significant capability. The digital signature is a kind of signature that serves two primary purposes: firstly, to ensure that the message has been sent to the intended recipient without any alterations, and secondly, to offer assurance on the authenticity of the sender's identity. RSA algorithm is often used for the safe transmission of cryptographic keys over vulnerable communication channels. The method employs two keys because of its asymmetric nature. There are two distinct key types, public key pk and private key sk. In the cryptosystem, key pk is readily available to all individuals, while key sk is safeguarded in secrecy by authorised individuals. The RSA encryption algorithm is widely used within the electronic sector for facilitating secure online financial transactions. The detailed algorithms are shown below [5]:

Key generation:

Pick two random prime d and e, compute n = d * e and $\emptyset(n) = (d-1)(e-1)$.

Choose an integer $g \in [1,\ \emptyset(n)-1]$ randomly, and calculate its inverse $h\, mod\, \emptyset(n)$.

$sk = (h, n),\ pk = (g, n)$.

Encryption:

Pick the message m.

Compute the ciphertext $c = m^g\, mod\, n$.

Decryption:

Receive the ciphertext c.

Compute the message $m = c^h\, mod\, n$.

2.2. ElGamal

ElGamal is another widely used public-key algorithm which was first developed in 1985 by Taher ElGamal [6]. Compare with RSA, ElGamal is based on a different computationally hard problem and Diffie-Hellman key exchange [3]. It serves as an alternative to the RSA method for the purpose of public key encryption. ElGamal algorithm can be implemented for encryption and digital signature creation. The detailed algorithm for the key generation, encryption and decryption are shown below [7]:

Key generation:

Choose the cyclic group G and determine its generator g and group prime order p.

Choose value $x \in Z_p$
Compute $y = g^x$.
sk = x, pk = y.
Encryption:
Pick random value $x \in Z_p$.
Pick the message m.
Compute the ciphertext $c = (c_1, c_2) = (g^r, y^r \cdot m)$.
Decryption:
Receive the ciphertext c.
Compute the $m = c_2 \cdot (c_1)^{-x}$

2.3. *Elliptic Curve Cryptograph (ECC)*

The introduction of ECC was separately done by Neal Koblitz and Victor Miller in 1987 [8]. ECC is a cryptographic technique that converts a mathematical issue into a computer method using the finite field elliptic curves. To represent an elliptic curve, consider it is over a group F with a prime order q, the curve can be represented by the following equation [9]:

$$y^2 = x^3 + \mathrm{ax} + b \tag{1}$$

The value of x and y are shown as a point on the curve if they satisfy the equation. The collection of points on a curve is augmented by a unique point known as the point at infinity. Let A and d be a point and positive integer. The scalar point multiplication, which has the form d*A, is a fundamental component of all elliptic curve cryptosystems.

If two parties want to use ECC, each party will choose a random integer as the private key k_1, k_2. After that, they can compute their public keys by multiply a common point P with their private keys which are k_1P, k_2P. Therefore, they can share the same secret key by multiplying other side's public key with their own private key, which are k_2k_1P, k_1k_2P . This process is also called $Elliptic\ Curve\ Diffie\ Hellman\ \ key\ exchange\ (ECDH)$. Another widely used ECC application is $elliptic\ curve\ digital\ signature\ algorithm\ (ECDSA)$ that can be used for the sign and verify processes in the cryptocurrencies like Bitcoin [10].

3. Underlying Mathematical Problems

In the asymmetric cryptography, the function that used in the algorithms should be the trapdoor function which should be evaluated easily and inverted difficultly without the specific information. The function should be the one-way unless the additional information is provided. The reliability of asymmetric cryptography algorithm is dependent on the computationally hard assumptions. The tasks for deducing the private key from associated public information are considered to be as challenging as solving a computationally hard problem. RSA, ElGamal and ECC are based on the different mathematical problems.

3.1. *Integer Factorization Problem*

The reliability of RSA algorithm depends on the computational difficulty of the integer factorization problem (IFP). Suppose there is a value $n = pq$ which is computed by multiplying p and q. Value p and q should be unknown primes. The hard problem is defined as the problem of finding p and q. The difficulty of RSA problem should be also dependent on above problem. If there is an adversary A can solve IFP in polynomial time, they can break RSA as well. The adversary can use A to compute p and q first, then find the $\emptyset(n)$ and private key. Finally, they can perform the decryption process. Therefore, the difficulty of IFP plays the essential role in security of RSA public cryptosystem.

3.2. *Discrete Logarithm Problem*

In 1976, Diffie and Hellman first introduced the $discrete\ logarithm\ problem$ and used in the asymmetric cryptosystem. The reliability of ElGamal depends on DLP. Let G = <g> be a $cyclic\ group$, and a value e is in the range of group order and value d is in the group, they satisfy $g^e = d$. The DLP is finding e for given group elements g, d such that $g^e = d$. If DLP can be solved by an adversary A in polynomial time, the adversary can use A to calculate the value e in polynomial time, which represent the private key. Hence, the computational complexity associated with DLP is crucial for the security of ElGamal cryptosystem.

3.3. *Elliptic Curve Discrete Logarithm Problem*

$Elliptic\ curve\ discrete\ logarithm\ problem\ (ECDLP)$ is an alternative version of DLP. The difficulty of alternative version problem is essential for the reliability of ECC. ECDLP is a computationally hard problem in the field of cryptography:

Assume there is a new group G that represents the collection of elliptic curve points, a value x is in the range of group order, E and D are points in the group, they satisfy $E = xD$. This problem is determining the value x from the values of E and D.

Above problem serves as the fundamental component used in elliptic curve key generation. The integer x serves as the private key, while the point E functions as the public key.

4. Comparative Analysis

For the key size, compared with RSA and ElGamal, ECC has the smallest key size in each security level shown in the table 1. This feature is also one of the primary benefits of elliptic curve cryptography (ECC) in comparison to other asymmetric cryptosystems.

Table 1. Key sizes (bits) of three algorithms in each level.

Security level	RSA/EIGamal	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

An additional crucial aspect to assess the performance is the examination of the duration required for key creation, encryption, and decryption processes. In general, the ECC algorithm exhibits the shortest running time among the three procedures [11, 12]. In the decryption process, the disparity in running time between ECC and the other two algorithms is much greater compared to the encryption and key generation phases. As the volume of processed data rises, the difference in execution time across the three algorithms will become more apparent. Therefore, ECC is characterised by its use of tiny keys and high computational efficiency, rendering it well-suited for contemporary devices with limited processing capabilities, such as smart cards and IoT devices. The use of ECC has emerged as the preferred cryptographic method for networks and communication devices. Furthermore, it should be noted that the ruling time of the ElGamal encryption scheme is comparatively lower than that of RSA, because ElGamal technique produces ciphertext that is more complex and exhibits a slower computational performance due to the generation of multiple public keys throughout the encryption and decryption processes. One feature of this encryption method is its ability to produce distinct ciphertexts for the same plaintext throughout each encryption round.

5. Conclusion

This study examines the efficacy of three asymmetric encryption techniques, RSA, El-Gamal and ECC. The cryptographic systems were evaluated based on the size of key and the running time for the key generation, encryption and decryption processes on the same security level. Comparisons demonstrate that the expense associated with transmission experiences a significant reduction in the context of Elliptic Curve Cryptography (ECC). Additionally, the findings also demonstrate the efficacy of ECC in practical applications. In future, a potential direction for further development is involving symmetric schemes to obtain more comprehensive outcomes.

References

[1] Ferguson, N., Schneier, B., & Kohno, T. (2011). Cryptography engineering: design principles and practical applications. John Wiley & Sons.

[2] Goyal, S. (2012). A Survey on the Applications of Cryptography. International Journal of Science and Technology, 1(3).

[3] Diffie, W., & Hellman, M. E. (2022). New directions in cryptography. In Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman (pp. 365-390).

[4] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[5] Milanov, E. (2009). The RSA algorithm. RSA laboratories, 1-11.

[6] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4), 469-472.

[7] Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.

[8] Miller, V. S. (1985, August). Use of elliptic curves in cryptography. In Conference on the theory and application of cryptographic techniques (pp. 417-426). Berlin, Heidelberg: Springer Berlin Heidelberg.

[9] Hankerson, D., & Menezes, A. (2021). Elliptic curve cryptography. In Encyclopedia of Cryptography, Security and Privacy (pp. 1-2). Berlin, Heidelberg: Springer Berlin Heidelberg.

[10] Koblitz, N., Menezes, A., & Vanstone, S. (2000). The state of elliptic curve cryptography. Designs, codes and cryptography, 19, 173-193.

[11] Sann, Z., Soe, T. T., & Nwe, K. M. (2019). Comparison of Public Key Cryptography in Different Security Level. International Journal of Recent Development in Engineering and Technology, 8(12).

[12] Maqsood, F., Ahmed, M., Ali, M. M., & Shah, M. A. (2017). Cryptography: A comparative analysis for modern techniques. International Journal of Advanced Computer Science and Applications, 8(6).