International Journal on Optimization and Applications

IJOA. Vol. 4, Issue No. 3, Year 2024, http://www.ijoa.ma/

Security analysis of RSA algorithm: vulnerabilities and countermeasures

1st Mina Zouhal
Dept. Informatique
ENSAK KENITRA,Morocc
mina.zouhal@uit.ca.ma

2nd Sihame Bacime
Dept. Informatique
ENSAK KENITRA,Morocc
sihame.bacime@uit.ca.ma

3rd Kaoutar EL hachimi
Dept. Informatique
ENSAK KENITA,Morocco
kaoutar.elhachimi@uit.ca.ma

Abstract—Cryptography is divided into two types, namely symmetric cryptography and asymmetric cryptography. In asymmetric cryptography, the encryption and decryption processes have their keys. This article provides a clear overview of the RSA algorithm's security vulnerabilities, with a specific focus on issues related to key management. It highlights the critical role that secure key management in RSA implementations by looking at issues including weaknesses during key distribution, unsafe storage techniques, and faults in key creation. The paper highlights the dangers associated with compromised keys and looks at doable methods to improve key security, such as strong encryption techniques and stringent access control protocols. The purpose of this article is to clarify the significance of proactive key management procedures in supporting the overall security posture of cryptographic systems based on RSA.

KEYWORDS—RSA, ECC, GF(2^m), Cryptography System, Hybrid System, Combination between RSA and ECC, Cooperation RSA with ECC

I. Introduction

In the realm of digital communication and encryption, the RSA algorithm, created by three founders (Ronald, Shamir, and Adleman) in 1977, is included in asymmetric encryption, which means it operates with two keys: one for encryption called the public key, and one for decryption called the private key. It is used in E-commerce, and also in confidential data exchange. The RSA algorithm protects data transmission, is a secure algorithm, and can be used in digital signatures (using two keys), and also in key exchange. RSA keys are typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future.

Despite these advantages, there are some of problems that can limit the use of RSA in some cases, specifically**,** the size of keys in the RSA system, which is critical to ensuring communication Security. Keys that are too small make encryption vulnerable to brute force and factorization attacks, weakening security levels. With technological advancements, attackers can exploit advanced computing capabilities to accelerate the process of breaking RSA keys. To ensure adequate security, it is recommended to use RSA keys with a minimum size of 2048 bits, following current security standards and anticipating technological advances**,** in our article we are going to discuss the problem of key length because the algorithm's strength depends on the key size. Thus RSA relies on the length of its keys to make them difficult to crack. We can summarize this by saying that" longer RSA keys are more secure and harder to hack than short ones.

We propose a solution, which is involves linking ECC with the RSA algorithm, to address the key size problem in RSA and highlight some of the hurdles that need to be overcome for these solutions to be successful. By the end of this essay, readers will have a better understanding of the solution regarding the key size in RSA, as well as the difficulties and opportunities associated with its implementation.

II. Algorithm RSA

A. **Definition** :

The RSA algorithm Known by the names of its three creators, Rivest, Shamir, and Adleman, the RSA algorithm is a popular asymmetric cryptographic for safe data transfer. It encrypts and decrypts data using a public key and a private key. Every participant in RSA creates a pair of keys: a private key that needs to be kept private and a public key that may be shared freely.

Usually, encryption uses the public key while decryption uses the private key.

RSA can be used in a variety of cryptographic applications, including key exchange protocols, digital signatures, and secure communication. Large integer factorization is hard, which is the foundation of its cryptographic strength and ensures security. In this article we will explain how RSA does it work, we will mention an essential part which is security of RSA his vulnerabilities and how can we solve this vulnerabilities.

B. How does the RSA work?

1. Generating -private key pair in RSA:

The process of creating a secure public-private key pair in RSA requires multiple steps that need to be taken:

i. Selection of Prime Numbers:

The first step in key generation is to select two distinct prime numbers, typically denoted as **p** and **q**. These primes are chosen to be large enough to resist factorization attacks.

ii. Computation of RSA Modulus:

Once the prime numbers **p** and **q** are selected, the RSA modulus, **n**, is computed as the product of these primes: **n = p * q**. The modulus **n** serves as the core parameter of the RSA algorithm and is included in both the public and private keys. The security of RSA dependent in the computational complexity of factoring **n** back to its main components, **p** and **q**.

iii. Compute Euler's totient function $\phi(n)$:

$\phi(n)=(p-1)\times(q-1)$. This function gives the number of positive integers less than **n** that are relatively prime to **n**.

iv. Selection of Public Exponent:

After calculating **n**, a public exponent, often denoted as **e**, is chosen. This exponent must be co-prime to $\phi(n)$ and $1<e<\phi(n)$, ensuring that it does not share any factors with **(p-1)(q-1)** except for 1. A common choice for the public exponent is **65537 (2^16 + 1),** as it has desirable cryptographic properties and speeds up the encryption and decryption operations.

The public key consists of the modulus **n** and the public exponent **e**. It is used for encryption **(n, e).**

The private exponent, **d**, is calculated such that $d \times e \equiv 1 \pmod{\phi(n)}$. In other words, **d** is the modular multiplicative inverse of **e modulo** $\phi(n)$.

The private key consists of the modulus **n** and the private exponent **d**. It is used for decryption **(d,n).**

2. Key Distribution:

RSA eliminates the need for a secure channel for key exchange, a requirement in symmetric key algorithms, by employing asymmetric encryption. Users only need to share their public keys openly. Recipients use their private keys for decryption. This approach simplifies key distribution, making it suitable for various applications. The public key **(n, e)** is openly distributed, while the private key **(d,n)** is kept secret.

3. Encryption and Decryption:

Encryption:

To encrypt a message **M** using RSA, the sender obtains the recipient's public **key (N, e)**. The plaintext message **M**, represented as an integer smaller than **n** $(0 \le m < n)$

undergoes modular exponentiation with the public exponent **e**: $C \equiv M^e \pmod{N}$.

The resulting ciphertext, C, is transmitted securely to the recipient.

Decryption:

Upon receiving the ciphertext **C**, the recipient applies their private key **(d,n)** to recover the original message **M**. Decryption is performed using the equation: $M \equiv C^d \pmod{N}$.

The recipient can then retrieve the plaintext message **M** from the decrypted ciphertext.**[3]**

C. The advantages RSA

The RSA algorithm is used in cryptography for its Security and privacy benefits. We discuss some important advantages of the RSA algorithm against vulnerabilities and countermeasures.

Even if factorization techniques are advancing quickly, modern data encryption systems will stay secure as long as it is difficult to break down numbers longer than 100 digits into first form. We begin by listing our advantages.

Resistance to factorization attacks : One of the main advantages of RSA is its resistance to factorization of large primes. Attacks aimed at factoring prime numbers used in RSA, such as Lenstra's general polynomial number factorization algorithm, are only effective for relatively Small key sizes. To counter this, using RSA keys of sufficiently large size makes these attacks impraticable.

To explain this additional advantage based on the resistance of RSA to factoring attacks, by highlighting the underlying mathematical issues which represent in the Complexity of factoring large primes : RSA is based on the principle that factoring a large number into products of primes is a difficult problem. More precisely, the RSA algorithm exploits the difficulty in factoring the product of two large prime numbers to find the original prime numbers. This difficulty is due to the lack of efficient factorization algorithms for large numbers.

RSA key size : The security of RSA depends largely on the size of the prime numbers used to generate the keys. To resist factorization attacks, RSA keys used in modern cryptographic systems are typically 2048 bits or longer. This makes factorization extremely difficult and requires massive computational resources, even with the most advanced algorithms.

Complexity of alternative methods : In addition to the general factorization algorithm, other methods have been developed to attack RSA, such as the quadratic sieve method and the general sieve method. However, all of these methods face similar difficulties when faced with the size of modern RSA keys.

En follows RSA provides significant protection against brute force attacks due to the difficulty of calculating inverse modular exponentiation, which is the basis of the algorithm. The length of RSA keys can be adjusted to make these attacks ineffective. Standard sized RSA keys, like 2048 bits, require considerable computational resources to brute force break.

RSA is Resistance to side-channel attacks : Side-channel attacks aim to exploit physical or temporal information, such as energy consumption or calculation time, to compromise the security of a cryptographic algorithm. RSA, when properly implemented, is less vulnerable to these types of attacks compared to other algorithms based on elliptic curves or symmetric algorithms.

These benefits make RSA a popular choice for securing online communications and transactions, but it is essential to consider implementation best practices to maximize its security.

D. RSA LIMITS

Because of its reliability and effectiveness, the RSA algorithm is frequently used in the field of cryptography. It is worth looking into for safe and efficient use, though, as it has certain drawbacks and crucial factors.

Key size: One of the most obvious limitations of RSA is key size. To ensure an adequate level of security, RSA keys must have sufficient length, generally expressed in bits. With the evolution of computer computing power and brute force attacks, it is necessary to use keys large enough to resist attacks.

Computational complexity: RSA relies on complex mathematical operations, including the factorization of large prime numbers. This complexity can make the key generation and encryption/decryption process expensive in terms of computing resources, especially for large amounts of data.

Potential Vulnerabilities: Although RSA is considered secure if properly implemented with appropriate settings, it may have vulnerabilities if errors are made in key generation, use of weak padding algorithms, or other aspects of implementation.

Side-Channel Attacks: Side-channel attacks, such as power analysis or timing attacks, can pose a threat to the security of RSA by exploiting information about the cryptographic operations themselves, rather than Aim directly at keys or encrypted messages.

Key Management: Key management in a system using RSA is crucial. Proper key management practices, such as regular key rotation, adequate protection of private keys, and revocation of compromised keys, are essential to maintaining system security.

Message size: RSA has encryption capacity limited by the size of the keys used. It is typically used to encrypt relatively short data, such as session keys in secure communications protocols, due to its complexity and limited performance for large volumes of data.

Evolution of Attacks: With the constant evolution of attack techniques and computing technologies, researchers sometimes discover new vulnerabilities or more effective attack methods against algorithms like RSA. It is therefore important to stay up to date on advances incrypt analysis and computer security.

III. Algorithmes ECC

A. Definition:

An asymmetric cryptography technique based on the characteristics of elliptic curves defined over finite fields or real numbers is called the Elliptic Curve Cryptography (ECC) algorithm.

The discrete logarithm problem's difficulty on an elliptic curve is the foundation of the ECC algorithm. It performs cryptographic computations by applying mathematical operations on points on the curve.

More precisely, a user selects a starting point on an elliptic curve and uses a sequence of steps known as "scalar multiplications" to produce a public key from their private key in order to create a key pair (public/private). The public key is the product of multiplying the starting point by the private key, which is a random number.

Compared to other asymmetric cryptography techniques like RSA, the ECC algorithm has a number of advantages, such as a smaller key size for comparable security. This makes it especially appropriate for settings like embedded systems and mobile devices where storage capacity and bandwidth are limited.

The ECC algorithm is widely used in many security protocols, such as communication encryption, digital signatures, because of its mathematical complexity and security robustness.**[1]**

B. How does the ECC work ?

1. ***Point Generation.*** **:**

We will create the binary field GF(2^m) points by .applying an irreducible polynomial equation

$$y^2 + xy = x^3 + ax^2 + b$$

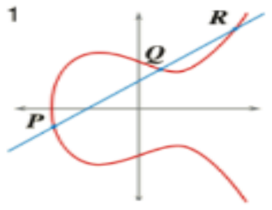
The binary field equation defines the elliptic curve over GF(2^m(field size)).

Where a and b are constants.

An elliptic curve will be formed by the union of .the generated points

2. **Elliptic curve :**

The Binary field Elliptic curve defined by the pairs (x,y) that satisfy the irreducible polynomial equation. P, Q, and R are the curve's points in this instance



$y^2+xy=x^3+ax^2+b$

3. *Key generation ECC*

The creation of a public and private key is a necessary step in the implementation of an ECC processor. The sender encrypts the message using the public key, and the receiver decrypts it using the private key.

The public key is produced using the following equation: Q=K.G, where k is the random number that represents the private key and is between (1 and n). Q is a public key G is a global parameter.

The steps of the key generation algorithm are listed below.

- *Choose the appropriate curve* **Eq(a,b)**
- *Choose a base point* **P = (x1,y1)** with large order **n**
- *Choose your private keys* s*uch that* $\mathbf{n_a < n}$ and $\mathbf{n_b < n}$

:Determine the public keys by computing

$P_a = n_a \,.\, P$ and $P_b = n_b \,.\, P$

4. ***Encryption*** **:**

A message is encoded during the encryption process so that only authorized parties can decipher it.

The equation that follows will be applied to encryption.

$$\mathbf{C_m = K*G, P_m + K*P_b}$$

where :

K is a randomly generated secret key.
C_m is a text that is encrypted.
P_m is the plaintext.
G is the generator point.
P_b is the public key.

5. **Decryption**

The process of decrypting involves returning the encrypted text to its original form.

The following equation will be applied to decryption.

$$\mathbf{P_m + K*P_b - K*G*n_b = p_m}$$

where :

K is a randomly generated secret key..
P_m is the plaintext.
G is the generator point.
P_b is the public key.
n_b is a secret key (private key of the receiver).**[2]**

C. ECC Advantages:

ECC(Elliptic Curve Cryptography) is a powerful encryption method with some key advantages.

Strong Security: Even with smaller keys than other methods(like RSA),ECC provides robust protection.This security relates to the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP),which is thought to be extremely hard to crack with computers.

Efficiency Boost: ECC is faster and requires less processing power than other methods. This makes it ideal for devices with limited resources, like mobile phones, internet-connected gadgets (IoT), and anything else that needs to save on battery and processing power.

Compact Keys: For the same level of security, ECC keys are much smaller than RSA keys. Smaller keys mean less storage space needed and faster encryption/decryption. This is great for situations where storage space is limited.

Quantum Computer Ready: Unlike some other methods, ECC is believed to be more resistant to attacks from future quantum computers. These powerful machines could break many current encryption methods, but ECC's math is thought to be more secure.

D. ECC LIMITS:

However, ECC also has some drawbacks:

Implementation Complexity: Setting up ECC requires a deep understanding of advanced math concepts. This can make it trickier to implement correctly compared to other methods. Mistakes during implementation can leave security holes.

Potential Patent Issues: Some ways of using ECC might be covered by patents. This could restrict its use or require fees for commercial applications. Developers need to consider these intellectual property issues when choosing ECC.

Performance can Vary: How efficient ECC is depends on the specific settings chosen. Different settings can impact speed and memory usage. Picking the right settings is important for optimal performance.

Key Management Challenges: Managing ECC keys, especially in large systems with many users and devices, can be complex. Secure generation, distribution, and storage of keys are crucial. Poor key management can leave a system vulnerable to attacks.

In conclusion, ECC offers significant benefits like strong security, efficiency, and resistance to quantum computers. However, it also has challenges in terms of complexity, patents, performance variations, and key management. Careful planning and expertise are needed to address these limitations and ensure successful ECC implementation.

IV. Proposed solution: combination ECC with RSA

Hybrid cryptography, combining ECC and RSA, offers the best of both worlds. ECC's efficiency tackles RSA's processing needs, while both provide robust security. This flexibility is crucial in today's complex cybersecurity landscape.

ECC shines as an innovative solution, addressing RSA's limitations. Its shorter keys reduce computational burden and improve scalability, especially in resource-constrained environments. Furthermore, ECC's resistance to quantum computing threats future-proofs encryption compared to RSA's vulnerability.

While RSA remains a cornerstone of cryptography, ECC's emergence represents a significant advancement. By leveraging ECC's compact keys, efficiency, and quantum resistance, we can strengthen existing RSA infrastructure, creating a more agile and resilient defense against evolving cyber threats.cryptography,ECC's emergence represents a significant advancement.

What are the differences between RSA and ECC?

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

The graph above demonstrates how ECC, with far less keys, may offer the same level of encryption strength as a system based on the RSA algorithm. A 256-bit ECC key, for instance, is equal to 3072-bit RSA keys, which are 50 percent larger than the 2048-bit keys that are in use at the moment. 128-bit keys are utilized by the most recent, more secure symmetric algorithms for TLS, such as AES. Thus, it makes perfect sense that asymmetric keys offer at least this degree of security.**[4]**

A. Key generation

1. ECC Algorithm:

Selecting a suitable elliptical curve should come first. The private key is then generated at random within the curve points' range. Next, multiply the private key by the curve's generator to determine the public key. ECC key creation is secure with this procedure.

2. RSA Algorithm:

Concerning the RSA algorithm: For every RSA key pair, a public key and a private key need to be generated. This process generates keys for an RSA cryptosystem.

B. Message encryption with AES:

Use the AES (Advanced Encryption Standard) algorithm to encrypt our message. AES is a symmetric encryption algorithm, meaning it uses the same key for encryption and decryption.

C. Signature of the message encrypted with ECC:

The encrypted communication can be signed using ECC. The validity and integrity of the message are ensured by the signature. Use the matching ECC public key to validate the signature after creating it with the ECC private key.

D. AES key encryption with RSA:

Use RSA to encrypt the previously created AES key because AES is a symmetric method. Due to the asymmetric nature of RSA encryption and the safe sharing of its public key, this enhances security.

E. Verifying the message signature with ECC:

Use the ECC public key to confirm the signature after obtaining the message and its signature. If the verification is successful, it verifies that the communication is authentic and originates from the intended source.

F. Decrypting AES key with RSA:

To decrypt the previously encrypted AES key, use the matching RSA private key. This enables us to decrypt the message using the original AES key.

G. Decryption of the message encrypted with AES:

Using the recovered AES key, decrypt the encrypted message. This allows you to recover the original message and make it readable.

REFERENCES

[1]https://cryptobook.nakov.com/asymmetric-key ciphers/elliptic-curve-cryptography-ecc
[2]Shantha A, Renita J and Edna Elizabeth N,<<Analysis and Implementation of ECC Algorithm in Lightweight Device>>, International Conference on Communication and Signal Processing, April 4-6, 2019, India
[3]https://ieeexplore.ieee.org/document/9002197
[4]https://www.globalsign.com/fr/blog/infos-sur-ecc-et-pourquoi-l-utiliser