Enhancing Elliptic Curve Digital Signature Algorithm (ECDSA) For the Implementation of Digitally Signed Emails

Mikaella Reign Gangoso[1], Ma. Corazon Macaraig[1]

[1]*Student, College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila (University of the City of Manila), Manila, Philippines*

Corresponding Author: gangoso.mikaella@gmail.com

***Abstract*: This study focuses on enhancing the Elliptic Curve Digital Signature Algorithm (ECDSA) for the implementation of digitally signed emails. ECDSA faces vulnerabilities due to weak elliptic parameters, random number generation flaws, and reliance on insecure hash functions. To address these issues, the study proposes a three-pronged approach: first, strengthening elliptic curve parameters to guard against brute force and invalid curve attacks; second, replacing random number generation with a deterministic nonce mechanism using SHA-256 to prevent private key leakage; and third, implementing SHA-256 as a secure hash function to resist collision attacks and improve signature integrity. These enhancements aim to create a more robust and reliable framework for securing email communications. The study is conducted in a simulated environment using Python, with a focus on ensuring practical improvements in ECDSA's security for academic settings.**

***Keywords*: ECDSA, digital signatures, email security, elliptic curve parameters, invalid curve attacks, deterministic nonce, SHA-256, private key protection, collision resistance, Python, cryptography.**

1. Introduction

This chapter provides the foundation for the study, examining the issues associated with using ECDSA for digitally signed emails. It highlights challenges such as weak elliptic parameters, private key leakage, and vulnerabilities from weak hash functions. The chapter outlines strategies to address these issues, discusses the research scope, beneficiaries, and limitations, and defines key terms.

Manuscript revised December 13, 2024; accepted December 10, 2024. Date of publication December 14, 2024.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898; SJIF: 5.59

Email communication is essential in academic environments, facilitating interactions between students and teachers regarding assignments, exams, and grades. Ensuring secure email communication through encryption and digital signatures is vital to protect sensitive academic data (Ghimire, 2023).

Digital signatures validate and authenticate electronic documents, ensuring integrity and non-repudiation in email transactions (Subramanya & Yi, 2006). The Elliptic Curve Digital Signature Algorithm (ECDSA) is a widely used method leveraging elliptic curve cryptography for efficient and secure key generation and signature verification, offering stronger security with smaller key sizes compared to RSA and DSA (Sanka et al., 2021).

Standardized in 1998 by ANSI, ECDSA is crucial for securing digital communications by ensuring authenticity, integrity, and non-repudiation. Its efficiency makes it suitable for constrained environments like university email systems (Kumar & Saadi, 2021). ECDSA operates through key generation, signature creation, and verification, allowing secure authentication without revealing private keys (Selikh, 2021; Khan et al., 2023).

Despite its advantages, ECDSA faces vulnerabilities such as signature reuse and weaknesses in random number generation, exposing systems to potential attacks (Puthiyidam et al., 2024). This study addresses these challenges by enhancing ECDSA's robustness to protect sensitive university email communications against emerging cyber threats.

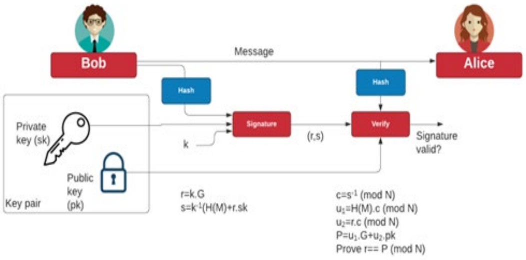
2. Methodology

The methodology for this project is structured around the Systems Development Life Cycle (SDLC) framework, employing the iterative model to ensure flexibility and continuous improvement throughout the development process. This approach enables the team to address evolving cryptographic security challenges by refining the system in successive iterations. The project begins with a comprehensive planning and requirement analysis phase, where key vulnerabilities, such as inadequate key and nonce management, faulty random number generation, and size overhead in ECDSA operations, are identified. A robust technology stack is chosen, including Python, PyCryptodome, and the ECDSA library, to provide strong support for cryptographic operations. Stakeholder interviews, system evaluations, and feasibility studies guide the design of the system, ensuring alignment with user needs and technical capabilities.

The design phase focuses on enhancing the ECDSA framework by introducing secure elliptic key pair generation, deterministic nonce creation using SHA-256, and the implementation of stronger hash functions to mitigate vulnerabilities like collision attacks. These enhancements are integrated into a digitally signed email system with modules for key management, email signing, and signature verification. The development phase involves coding the modules and integrating them into the existing or new email system infrastructure.

Thorough testing is conducted at multiple levels, including module, system-wide, and pilot testing, to ensure reliability and functionality. Feedback from the pilot implementation informs iterative refinements, enhancing usability, functionality, and security. This cycle of review, refinement, and reiteration ensures that the final system is robust, secure, and capable of addressing identified vulnerabilities while adapting to new cryptographic challenges.

3. ECDSA Framework



This streamlined framework focuses on the core elements of developing ECDSA for securing digitally signed emails, emphasizing key generation, process integrity, and continuous security enhancements.

A. Key Generation Algorithm

Used to generate the public and private key of the users.

1. Use an elliptic curve E over finite field Fp with p a prime number, and choose a point P ∈ E (Fp) which generates a cyclic group of a prime order n. Problem 1: Weak elliptic parameters make encryption vulnerable.

2. Choose a random integer d with $0 < d < n$.

3. Compute Q = dP.

4. The keys are now:
public key is Kpub = (p, n, Q, P).
private key is Kpr = d.

B. Signature Generation Algorithm

1. Select a random or pseudorandom integer k, $0 < k < n$. Problem 2: Vulnerability to Private Key Leakage Due to Faulty Random Number Generation.

2. Compute B = kP =(xB,yB).

3. Compute r ≡ xB mod n.

4. If r = 0 then go to step 1.

5. Compute hash value of the message h(m). Problem 3: Vulnerability of ECDSA to Weak Hash Functions and Collision Attacks.

6. Compute s ≡ [k−1(h(m)+dr)] mod n.

7. If s = 0 then go to step 1.

8. The signature of m is (r,s).

9. Sends the message m and the signature (r,s) to Bob.

4. Results And Discussion

The Modified Elliptic Curve Digital Signature Algorithm (ECDSA) introduces enhancements to improve security and robustness by addressing vulnerabilities inherent in traditional ECDSA implementations. The proposed modifications target three critical aspects: key generation, deterministic nonce generation, and the use of a more secure hash function.

A. Key Generation Algorithm:

The enhanced key generation algorithm ensures the use of strong elliptic curve cryptographic parameters. By employing elliptic curves over finite fields with a strong prime p > 2^256, it mitigates vulnerabilities associated with weak parameters. The selected point P on the curve generates a cyclic group of prime order n, with a small cofactor h, ideally set to 1. A private key d is randomly chosen within the range $0 < d < n$, and the corresponding public key Q is computed as Q = dP. This process ensures robust cryptographic strength, reducing susceptibility to invalid curve and brute force attacks. Simulations validate the efficacy of this approach, demonstrating significant reductions in attack frequencies when compared to traditional parameter sets.

B. Deterministic Nonce Generation:

The proposed enhancement addresses the risk of private key leakage caused by nonce reuse. Traditional ECDSA relies on random or pseudorandom integers for nonce k, which, if reused, exposes the private key. The modified approach deterministically generates k by hashing the combination of the message hash h(m) and the private key d using SHA-256. This process ensures k is unique for every message and private key pair, eliminating vulnerabilities tied to predictable or reused nonces. Extensive simulations across 10,000 iterations showed no nonce reuse or private key breaches, underscoring the reliability of the deterministic generation method.

C. Secure Hash Function Implementation:

To enhance resistance to collision and forgery attacks, the algorithm replaces weaker hash functions like MD5 and SHA-1 with SHA-256. This transition ensures cryptographic robustness by leveraging SHA-256's strong collision resistance, preimage resistance, and second preimage resistance properties. Simulations compared collision probabilities across hash functions, highlighting a substantial reduction in vulnerabilities when SHA-256 is employed. Visualization of collision counts further demonstrated the security advantages of the upgraded hashing mechanism.

Together, these enhancements significantly bolster the security of ECDSA, making it a robust solution for digitally signing emails and protecting sensitive communications. By addressing key vulnerabilities through a systematic and simulated approach, the modified ECDSA stands as a critical improvement over traditional implementations, ensuring the integrity and authenticity of digital signatures.

D. Conclusion

In the proposed system, we have implemented an organization-oriented system that would assist the human resource department in short listing the right candidate for a specific profile. The system could be used in many business sectors that will require expert candidates, thus reducing the workload of the human resource department.

5. Conclusion

This research successfully addresses key vulnerabilities within the Elliptic Curve Digital Signature Algorithm (ECDSA) through three targeted enhancements. First, the researchers developed a Python-based key generation mechanism that ensures adherence to strong elliptic curve cryptography parameters, specifically utilizing the NIST P-256 curve. This approach minimizes susceptibility to brute force and invalid curve attacks by securing cryptographic keys with adequate strength. Second, the researchers implemented a deterministic mechanism for nonce generation to prevent private key leakage. By generating unique nonces based on the private key and message hash, this method effectively mitigates risks associated with nonce reuse. Finally, the researchers enhanced ECDSA's resilience to forgery and signature manipulation by incorporating SHA-256 as the default hash function, reducing vulnerability to collision attacks. Simulations confirmed the efficacy of these enhancements, demonstrating a substantial reduction in attack incidences and reinforcing the security of digitally signed emails.

References

[1] Al-Zubaidie, M., Zhang, Z., & Zhang, J. (2017). Efficient and secure ECDSA algorithm and its applications: A survey. Ar5iv.

[2] Android Security Vulnerability. (2024). Bitcoin.org.

[3] Aranha, D., Rodrigues, F., Takahashi, A., Tibouchi, M., & Yarom, Y. (n.d.). LadderLeak: Breaking ECDSA With Less Than One Bit of Nonce Leakage.

[4] Barker, E., & Roginsky, A. (2019). Transitioning the use of cryptographic algorithms and key lengths. Transitioning the Use of Cryptographic Algorithms and Key Lengths.

[5] Bäumer, F., & Brinkmann, M. (2024). Understanding a critical vulnerability in PuTTY biased ECDSA nonce generation revealing NIST P-521 private keys (CVE-2024-31497) - vsociety. Vicarius.io.

[6] Biham, E., & Neumann, L. (n.d.). Breaking the Bluetooth Pairing - The Fixed Coordinate Invalid Curve Attack Workshop on Attacks in Cryptography 2.

[7] Breitner, J., & Heninger, N. (2019). Biased Nonce Sense: Lattice Attacks against Weak ECDSA Signatures in Cryptocurrencies. IACR Cryptology EPrint Archive;

[8] Erdogan, Y. (2023, March 21). The Elliptic Curve Digital Signature Algorithm ECDSA. DEV Community.

[9] Genc, Y. & Afacan, E. (2021). Design and Implementation of an Efficient Elliptic Curve Digital Signature Algorithm (ECDSA). 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS).

[10] Ghimire, N. (2023). Analyzing Email Communication Patterns Among Bachelor of Information and Communication Technology Education (BICTE) Students: A Case Study. Innovative Research Journal, 3(2), 55–66.

[11] Hussein, N. T., & Kashmar, A. H. (2020). An improvement of ECDSA weak randomness in blockchain. IOP Conference Series: Materials Science and Engineering, 928.

[12] Kaur, H., Sanghavi, J., Vakil, A., & Shah, S. (2024). A Study on Efficient Information Security using Elliptic Curves.

[13] Khalique, A., Singh, K., & Sood, S. (2010). Implementation of Elliptic Curve Digital Signature Algorithm. International Journal of Computer Applications, 2(2), 21–27.

[14] Khan, M., Kamal U., Alam, M., Khan, H., Shams Tabrez Siddiqui, Haque, M., & Parashar, J. (2023). Analysis of Elliptic Curve Cryptography & RSA. Journal of ICT Standardisation.

[15] Kumar, B., & Maiya Al Saadi. (2021, March 14). A Review on Elliptic Curve Cryptography. ResearchGate; unknown.