

## **Enhancing financial transaction security with lightweight cryptographic algorithms**

Vivek Deshpande \*

*Department of Computer Engineering  
Vishwakarma Institute of Information Technology  
Pune  
Maharashtra  
India*

Deepti Khubalkar <sup>†</sup>

*Faculty of Law  
Symbiosis Law School  
Nagpur Campus  
Symbiosis International (Deemed University)  
Pune  
Maharashtra  
India*

Anishkumar Dhablia <sup>§</sup>

*Engineering Manager  
Altimetrik India Pvt Ltd  
Pune  
Maharashtra  
India*

M. Jahir Pasha <sup>‡</sup>

*Department of Computer Science and Engineering  
G. Pullaiah College of Engineering and Technology  
Kurnool  
Andhra Pradesh  
India*

---

\* E-mail: [vivek.deshpande@viit.ac.in](mailto:vivek.deshpande@viit.ac.in) (Corresponding Author)

<sup>†</sup> E-mail: [deeptik@slnagpur.edu.in](mailto:deeptik@slnagpur.edu.in)

<sup>§</sup> E-mail: [anishdhablia@gmail.com](mailto:anishdhablia@gmail.com)

<sup>‡</sup> E-mail: [jahir4444@gmail.com](mailto:jahir4444@gmail.com)

Dharmesh Dhabliya <sup>®</sup>

*Department of Information Technology  
Vishwakarma Institute of Information Technology  
Pune  
Maharashtra  
India*

Yatin Gandhi <sup>#</sup>

*Competent Softwares  
Pune  
Maharashtra  
India*

---

## Abstract

The lightweight cryptographic algorithms used to increase the safety for financial transactions. The proposed lightweight cryptographic algorithms provide a middle ground, guaranteeing strong security with low computational overhead. Lightweight cryptography approaches being investigated for their potential in protecting private financial information during transactions. These algorithms are well-suited for real-time monetary transactions since they not only strengthen data integrity but also cut processing times through streamlined encryption and decryption processes. This paper focus into the planning, execution, and efficiency assessment of such lightweight cryptographic algorithms as they pertain to monetary systems. This study shows how successful they are at preventing common security threats like eavesdropping and data manipulation through extensive testing and analysis. Lightweight cryptographic algorithms have been shown to have the ability to dramatically improve the security of financial transactions, providing an efficient and cost-effective means of safeguarding sensitive financial data in today interconnected and potentially dangerous digital environment.

---

**Subject Classification:** 68M25.

**Keywords:** *Lightweight cryptography, Financial transaction, Chaotic map algorithm, Lightweight ECC.*

## 1. Introduction

The security of financial data has become a critical concern in an era marked by the growing digitization of financial systems and the growth of online transactions. It is imperative that the financial institutions, as well as the individuals and enterprises involved in these transactions, guarantee

---

<sup>®</sup> E-mail: [dharmesh.dhabliya@viit.ac.in](mailto:dharmesh.dhabliya@viit.ac.in)

<sup>#</sup> E-mail: [gyatin33@gmail.com](mailto:gyatin33@gmail.com)

the confidentiality, integrity, and validity of sensitive financial information. This calls for the investigation of novel strategies to improve the security of financial transactions. The algorithms classified as lightweight cryptography emphasize efficiency, speed, and resource optimization first, which makes them especially useful in situations when computational resources are scarce. Adopting lightweight cryptographic algorithms can give a balanced approach in the context of financial transactions, providing strong protection without imposing undue processing overhead.

In the past, to protect financial transactions, financial institutions have depended on well-known cryptographic standards like Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), and RSA. Although these techniques offer a high degree of security, they could be computationally demanding, which could lead to processing delays. Due to its capacity to provide security with the least amount of computing overhead, lightweight cryptography has drawn interest. A flexible and adaptable approach to security is required given the growing ubiquity of digital currencies, online payment methods, and mobile banking. Even though they are extremely secure, traditional cryptographic techniques can be computationally demanding, which could cause delays and inefficiencies when processing transactions in real time.

2. Proposed Method

The suggested approach for carrying out safe third-party financial transactions using cloud computing is made to lower the expenses associated with encryption and decryption while also improving the security of these transactions.

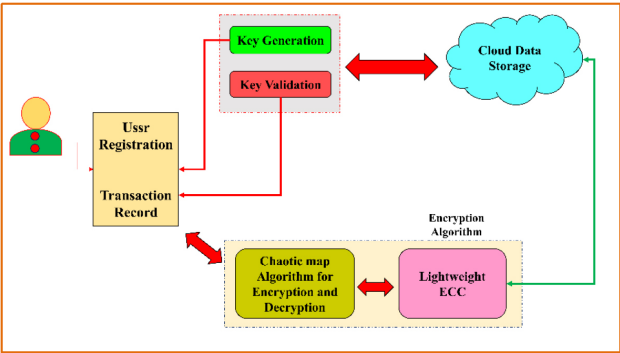


Figure 1  
Proposed system Architecture

The secrecy, integrity, and authenticity of transaction records are guaranteed by this approach, which makes use of multi-factor authentication and blockchain technology. The architecture is divided into multiple important phases, as shown in Figure 1, all of which play a vital role in enhancing the overall security and effectiveness of third-party financial transactions inside the cloud ecosystem. The suggested method's initial step entails strong user authentication. Before being granted access to the system, users including outside parties must go through multi-factor authentication.

### 2.1 Initialization phase

Ensuring the security and validity of financial transactions is a vital step in the startup process, which focuses on authorization and authenticity. This stage consists of a set of actions that authenticate the user or party starting the transaction and give them permission to carry out the planned financial activity. When two or more authentication elements are needed for user verification, Multi-Factor Authentication (MFA) improves security. Through the use of Fine-Grained Access Control, Policy-Based Access Control, or Role-Based Access Control (RBAC), authorization grants user permissions.

### 2.2 Key generation phase

The 128-bit cypher key, known as “Kc,” is first encrypted by dividing it into two equal parts, each consisting of 64 bits. These 64-bit segments are then split into 4-bit sub segments in the next phase. Each of these 4-bit sub segments is subjected to a shift operation, and the shifted outputs that are produced are then fed into a function called the “f-function.”

As the f-function functions on 4 segments, each of which has 4 bits (16 bits total). It uses equation (1) to produce a replacement for the cypher key (Kc):

$$A = B + C = 1Kc \ 4(j-1) + i \quad (1)$$

In this case, “i” equals 5, and “Kai<sup>f</sup>” is the result of solving this equation. Equation (2) yields this “Kai<sup>f</sup>” outcome.

$$v(x) = c(y) \quad (2)$$

The “f” function, consists of linear and non-linear transformations incorporating confusion and diffusion components. Every time the f-function is applied, a 16-bit 4x4 output matrix is created. In this process,

eight such arrays are formed, designated as 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'M7', and 'M8'. The key matrix is made up of all of these arrays.

Then, the key matrix arrays are converted into round keys, which are represented by the letters "K1, K2, K3, K4, K5, K6, K7, and K8." What emerges from the rotation of these arrays is "KM1, KM2, KM3, KM4, KM5, KM6, KM7, and KM8." The four public keys are derived using these generated key combinations in the following ways: " $kk1 = k1 + k2$ " and " $kk2 = k3 + k4$ ."

### *2.3 Authentication phase*

- User Request: When a user makes a request to access a system or service, the authentication process starts.
- User Identification: A username or other distinctive identifier is usually used by the system to identify the user.
- Authentication Server: For verification, the system forwards the authentication data to an authentication server.
- Access Authorization: Upon verification of the user's identification, access is authorized by the system according to their degree of authorization.
- Establishment of Session: After a successful authentication attempt, the user's session is created, enabling them to communicate with the system or service.

### *2.4 Encryption and Decryption*

Encryption, which comes after sub-key generation (KK1, KK2, KKK), is essential to protecting data while it's being transmitted or stored. The goal of this procedure is to produce a cryptographic alteration that makes the original material extremely difficult for unauthorized parties to access. During the encryption process, a number of basic operations are carried out, such as XOR, XNOR, AND, OR, left shift (LS), swapping, and substitution utilizing S-boxes. By adding confusion and diffusion to the data, these actions together improve its security in the following ways:

- Left Shift (LS): In a binary format, left shifting is the bit-level action of shifting bits to the left. It further obscures the structure of the data by assisting in distributing the impact of particular bits across several places.

- Substitution (S-boxes): Predefined values are substituted for particular bit patterns using substitution procedures. Tables or functions that carry out these replacements are called S-boxes. By modifying the data in a non-linear and intricate way, they create both confusion and diffusion while fortifying it against attacks such as differential and linear cryptanalysis.

Together, these processes make the encrypted data more complicated, making it more difficult for outside parties to decode the original data without knowing the encryption key.

### 3. Lightweight ECC Algorithm for Encryption and Decryption

#### *LECC Algorithm*

1. Initialize ECC parameters:
  - Choose an elliptic curve (E) and a base point (G) on the curve.
  - Define the order (n) of the base point G.
2. Key Generation:
  - Generate a private key (k) as a random number within the range [1, n-1].
  - Compute the public key (K) as  $K = k * G$ .
3. Transaction Encryption (Sender):
  - Create a shared secret (S) by performing scalar multiplication:  $S = k * \text{Receiver's Public Key}$ .
  - Derive a symmetric encryption key (KEK) from S using a key derivation function (KDF).
  - Encrypt the transaction details using KEK and a secure symmetric encryption algorithm (e.g., AES-GCM).
4. Transaction Decryption (Receiver):
  - Compute the shared secret (S) by performing scalar multiplication:  $S = k * \text{Sender's Public Key}$ .
  - Derive the same symmetric encryption key (KEK) from S using the same key derivation function (KDF).
  - Decrypt the transaction details using KEK and the same encryption algorithm.

5. Transaction Verification:
- Ensure that the sender’s public key and transaction details match the signed transaction data.
  - Verify the authenticity and integrity of the transaction using digital signatures.

4. Chaotic map Algorithm

By adding complexity and unpredictability, the chaotic map algorithm is integrated into the multifactor authentication system to improve security.

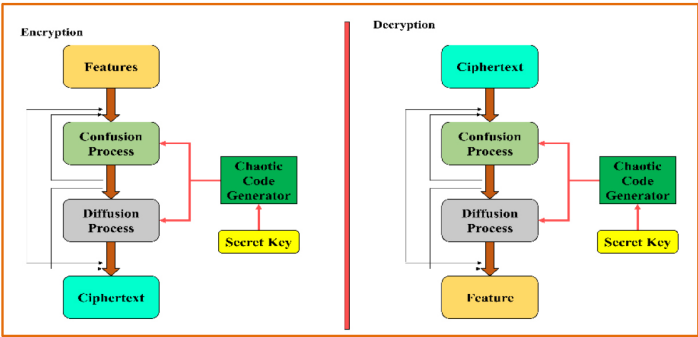


Figure 2  
Representation of Chaotic map Process for Confusion and Diffusion

Chaotic maps as shown in Figure 2 are used in this procedure to generate and validate the second component of authentication, which is usually a biometric identification. Any change in the hash value following an initial first factor authentication initiates the second factor, which involves transforming biometric data through chaotic map-based confusion and diffusion.

5. Performance Analysis

Table 1 compares the encryption computation of lightweight Elliptic Curve Cryptography (ECC) on different transaction data sizes. The sizes of the data vary from 200 KB to 1200 MB. ECC encryption takes a respectable amount of time on average, as evidenced by the time it takes to encrypt these various data quantities. The average encryption time for the highest file size of 1200 MB is about 1.046 seconds, showing that ECC

functions effectively even with large data volumes. Small to medium-sized data are especially well-suited to ECC, since the average encryption duration decreases with file size.

**Table 1**  
**Lightweight ECC for Encryption and Decryption computation on different size of data**

| Encryption | Transaction File Size | 50 MB | 200 KB | 500 KB  | 800 KB | 1200 MB |
|------------|-----------------------|-------|--------|---------|--------|---------|
|            | Time Taken (s)        | 2.6   | 1.008  | 1.016   | 1.3    | 1.03    |
|            |                       | 3.11  | 1.01   | 1.021   | 1.4    | 1.04    |
|            |                       | 3.6   | 1.013  | 1.026   | 1.5    | 1.053   |
|            |                       | 4.058 | 1.015  | 1.03    | 1.62   | 1.061   |
|            | Avg time taken (s)    | 3.342 | 1.0115 | 1.02325 | 1.455  | 1.046   |
| Decryption | Time Taken (s)        | 1.61  | 1.018  | 1.026   | 1.31   | 1.04    |
|            |                       | 2.32  | 1.02   | 1.031   | 1.41   | 1.05    |
|            |                       | 3.23  | 1.023  | 1.036   | 1.51   | 1.063   |
|            |                       | 3.058 | 1.025  | 1.04    | 1.63   | 1.071   |
|            | Avg time taken (s)    | 3.352 | 1.0215 | 1.03325 | 1.465  | 1.056   |

A comparison of the processing times of a few symmetric algorithms, such as DES, 3DES, AES, and LECC, with lightweight ECC is given in Table 2, Figure 3 and 4. ECC is a public-key cryptography technique that performs competitively. ECC encryption and decryption speeds are

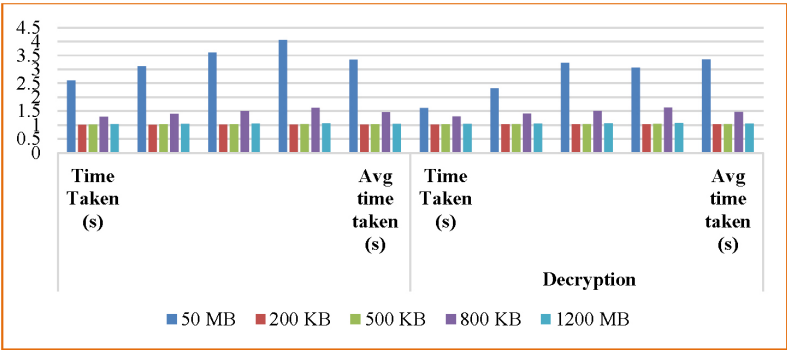


**Figure 3**  
**Representation of Lightweight ECC for Encryption and Decryption computation**

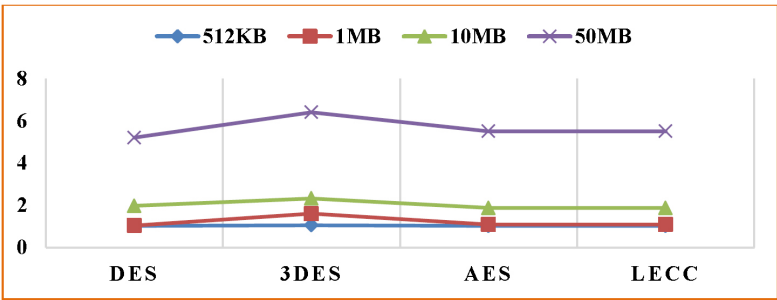


**Figure 4**  
**Processing time comparison for some Symmetric Algorithm with Lightweight ECC**

similar to those of conventional symmetric techniques like DES and AES for file sizes ranging from 512 KB to 50 MB. This implies that ECC can provide strong security at a processing efficiency as shown in Table 3 comparable to well-known encryption techniques. ECC also offers the benefit of having shorter key lengths for similar security, which lowers computational cost and makes it ideal for contexts with limited resources.

**Table 3**  
**Processing time comparison for some Symmetric Algorithm with Lightweight ECC**

| Size  | DES  | 3DES  | AES   | LECC  |
|-------|------|-------|-------|-------|
| 512KB | 1.02 | 1.054 | 1.022 | 1.022 |
| 1MB   | 1.04 | 1.602 | 1.09  | 1.09  |
| 10MB  | 1.97 | 2.32  | 1.87  | 1.87  |
| 50MB  | 5.2  | 6.4   | 5.5   | 5.5   |

**6. Conclusion**

Securing sensitive data integrity and confidentiality has advanced significantly with the use of lightweight cryptographic methods in financial transaction security. This study has shown that efficient encryption techniques and lightweight cryptographic techniques like elliptic curve cryptography (ECC) can be used to strengthen the security of financial transactions. We achieve a trade-off between strong security and low computational overhead by using lightweight techniques, which is important for financial data applications. These algorithms are a good

fit for real-time transaction processing because they have demonstrated to be extremely efficient in terms of speed and resource usage. Additionally, they deal with the changing issues brought on by increasingly complex cyber threats. The security architecture is further strengthened by the integration of biometrics and secure key management with multi-factor authentication. These systems protect the cryptographic keys that are essential for safe transactions in addition to authenticating user identities.

## References

- [1] R. Yegireddi and R. K. Kumar, "A survey on conventional encryption algorithms of Cryptography," 2016 International Conference on ICT in Business Industry & Government (ICTBIG) (2016).
- [2] Sina Rafati Niya; Eryk Schiller; Burkhard Stiller, "Architectures for Blockchain-IoT Integration1," Communication Networks and Service Management in the Era of Artificial Intelligence and Machine Learning, IEEE, pp.321-344 (2021).
- [3] A. Sengupta, E. R. Kumar and N. P. Chandra, "Embedding Digital Signature Using Encrypted-Hashing for Protection of DSP Cores in CE," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 398-407, Aug. (2019).
- [4] K. Jarvinen and J. Skytta, "On Parallelization of High-Speed Processors for Elliptic Curve Cryptography," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 16, no. 9, pp. 1162-1175, Sept. (2008).
- [5] Muhammad Asad Saleem, SK Hafizul Islam, Shafiq Ahmed, Khalid Mahmood, Majid Hussain, "Provably secure biometric-based client-server secure communication over unreliable networks", *Journal of Information Security and Applications*, Vol. 58, 102769 (2021).
- [6] K. Aoki et al., "Camellia: a 128-Bit block cipher suitable for multiple platforms design and analysis," (2001).
- [7] M. Ebrahim and C.W. Chong, "Secure Force: a low-complexity cryptographic algorithm for Wireless Sensor Network (WSN)," (2013).
- [8] S. Purohit, V. Deshpande and V. Ingale, "FPGA Implementation of the AES Algorithm with Lightweight LFSR-Based Approach and Optimized Key Expansion," 2023 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA), Bangalore, India, pp. 1-7 (2023), doi: 10.1109/PKIA58446.2023.10262697.

- [9] Z. Ghaffar, S. Shamshad, K. Mahmood, M. S. Obaidat, S. Kumari and M. K. Khan, "A Lightweight and Efficient Remote Data Authentication Protocol Over Cloud Storage Environment," in *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 103-112, 1 Jan.-Feb. (2023).
- [10] X. Hao, W. Ren, Y. Fei, T. Zhu and K. -K. R. Choo, "A Blockchain-Based Cross-Domain and Autonomous Access Control Scheme for Internet of Things," in *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 773-786, 1 March-April (2023).
- [11] Y. Zheng, W. Liu, C. Gu and C. -H. Chang, "PUF-Based Mutual Authentication and Key Exchange Protocol for Peer-to-Peer IoT Applications," in *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 3299-3316, 1 July-Aug. (2023).
- [12] H. Yue, X. Wei, J. Tan, N. Jiang and M. Qiu, "Eff-ECC: Protecting GPGPUs Register File With a Unified Energy-Efficient ECC Mechanism," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 7, pp. 2080-2093, July (2022).
- [13] H. Sohn and R. H. Victora, "Recording Comparison of ECC versus Conventional Media at Equal Grain Size," in *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 4073-4076, Oct. (2011), doi: 10.1109/TMAG.2011.2157907.