Research Article

Design of a Secure and Efficient Electronic Payment System Tailored for Mobile Users

Neeraj Kumar[1], Prerna Mohit[2]

[1]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India. Email: 2401050032cse@gmail.com
[2]Amity School of Engineering & Technology, Amity University Hyderabad, Telangana, India. Email: prernamohit4@gmail.com

Abstract

The objective of the program is to create a new electronic payment scheme that is secure and efficient and appropriate for mobile phone subscribers. In the area of mobile payments, the framework addresses important issues such as security, efficiency, user privacy, and lightweight processing appropriate for devices with restricted resources. A multi-level security approach is followed by our protocol, which consists of temporary identities, RSA digital signatures, and symmetric key cryptography. All this is achieved while maintaining a reasonable level of communication overhead. Through the establishment of a key management system that has been carefully designed, the system proposed establishes secure paths between the client, the merchant, the payment gateway, the issuer, and ultimately the acquirer. By shortening transaction processing times by 37% and computing overhead by 42%on mobile devices, our solution provides better performance compared to existing frameworks, as our experimental results demonstrate. This is achieved while maintaining the security requirements that are mandated by the National Institute of Standards and Technology (NIST). With significant gains in transaction throughput, security guarantee, and resource usage, the implementation results indicate that the system is acceptable for deployment in real-world settings.

Keywords: Electronic payment, Mobile security, RSA signatures, Payment protocol, Cryptography, Transaction privacy, Authentication, Mobile commerce, Secure communication, Payment gateway.

1. Introduction

A revolution in electronic commerce has been initiated because of the exponential growth in the use of mobile devices. This revolution has resulted in a change in the way clients make financial transactions. Based on the latest information available from the industry, it is estimated that the total value of mobile payment transactions would amount to $ 4.7 trillion worldwide by the year 2025 [1]. This shows the critical necessity of electronic payment systems, which are not only secure, but also efficient, and are designed specifically to be employed in mobile environments. This is in contrast to the mobile payment systems that are actually being used, which are still plagued by a range of issues, including security vulnerabilities, high computational burdens placed on limited devices, and privacy issues. Mobile payment systems are bounded by issues that are unique to them, when compared to the types of electronic payment frameworks that are traditionally employed. The limited processing capability, the unreliability of the network connection, the limited battery life, and the inherent vulnerabilities, which are found with wireless communication channels, are just a few examples of the limitations that are present with wireless communication channels [2] [3]. Moreover, since mobile devices are inherently personal, there is a necessity for enhanced protection of privacy. This is because access to payment credentials by 9an individual has the potential to cause severe financial and personal harm. Current payment systems are heavily based on complicated cryptographic techniques that are resource intensive on the device, or centralized systems that create bottlenecks and single points of failure. Both are heavy burdens. Moreover, much of the current research fails in its management of the balance between security and efficiency. Consequently, this results in systems that are insecure or far too complicated to be practical for mobile usage. Both of these outcomes are undesirable. This research introduces a novel electronic payment mechanism that has been specifically built for usage in mobile settings. The purpose of this mechanism is to overcome the limits that have been found. This system that we have introduced utilizes a combination of both symmetric and asymmetric cryptographic techniques so that all entities taking part in this can communicate securely with each other at the same time minimizing the processing overhead. All this for the purpose of achieving both the following goals. The protocol's deployment creates a framework which is not only lightweight, but also resilient. This design does not require any extra resources of the cellular system and provides high security guarantees, reducing the risk of a security breach.

1. Literature Review

In the course of the last decade, there has been a significant shift in the research that is being conducted on secure mobile payment systems. This change has occurred for a number of reasons. It is at this very instant in time that this change has taken place. In order to classify the techniques that are being used, we use a categorization system. These strategies are arranged into four fundamental areas. Token-based systems, cryptographic protocol-based systems, biometric identification systems, and solutions that are established on blockchain technology are some of the many types of systems that may be divided into these categories. There are many more types of systems as well. Token-based payment systems are not only able to secure the identities of users, but they are also able to protect the financial information that users offer via their transactions. This is made feasible by the use of payment tokens that are ephemeral. When it comes to the recommendations that Zhang et al. [4] made in their study that was discussed before, this is in keeping with what they suggested. In spite of the fact that these systems would need a significant amount of server-side infrastructure and might possibly cause a delay in the processing of transactions, they guarantee to provide a high level of privacy protection once they are put into operation. That being said, this is in spite of the fact that they would be required. A method that is based on tokens was described by Li and Wang [5], who offered a description of the method. The use of this technology makes it possible to reduce the costs that are connected with communication; yet, at the same time, it is very dependent on implementation environments that can be depended upon. thus, the deployment flexibility inside the system is thus limited as a consequence of this consequence. These systems are built on cryptographic protocols, which serve as the foundation for these systems. These protocols are the foundation for those systems that put a focus on mathematical security guarantees via the use of encryption and digital signatures. These kinds of systems are often referred to as cryptographic systems depending on the context in which the word is used. Sharma and Kumar [6] proposed a method that provides a high level of security while using key sizes that are less than those that are typically required by RSA implementations. This method was able to do this by utilizing smaller key sizes. There is a particular kind of encryption known as elliptic curve cryptography (ECC), and it is used in order to lay the foundation for this technology. On the other hand, the execution of these apps requires a significant amount of processing power, which makes them less suitable for mobile devices that have capabilities that are on the lower end of the spectrum that is accessible compared to other mobile devices [7] [8] [9]. The outcomes of research that was carried out by Patel and colleagues [10] indicated amazing security characteristics; nonetheless, the study also projected that performance would suffer when there were substantial transaction loads present. Some similarities can be seen between this and the prior study that was also stated in the previous sentence. The design of biometric authentication systems, which are intended to concurrently enhance security, takes use of a combination of physiological or behavioural characteristics. This is done in order to simultaneously improve security. In order to prevent illegal access to persons, these systems are designed to provide protection. While Chen and Liu [6] included facial recognition into their payment system, Wang et al. [5] combined fingerprint authentication with transaction verification. This is in contrast to Chen and Liu's approach. When compared to Chen and Liu [11], this is in contrast. In order for them to achieve the goals that they had set for themselves, the action in issue was carried out. These technologies do enhance user authentication; however, they also increase the amount of work that has to be done on the computer and create privacy issues that are connected to the keeping and processing of biometric data from users [12]. Despite the fact that these technologies improve user authentication, they also bring privacy difficulties. Additionally, despite the fact that these technologies make user identification more efficient, they also present challenges with regard to privacy. It has been seen that there has been a significant amount of interest in payment systems that are based on blockchain technology over the course of the last several years. This interest has been observed. As a result of this observation, there has been a rise in interest. A basic technique for mobile payments that is based on blockchain technology was created by Garcia-Alfaro and colleagues [2] with the intention of lowering the amount of money that is necessary to establish a consensus. The purpose of this activity was to further improve the convenience of mobile payments in comparison to previous methods. However, on the other hand, their method continues to address issues that are connected to scalability and continues to make use of confirmation delays for transactions that are rather extensive. The initial effort [13] that Nakamoto made was the one that introduced the idea of blockchain technology to people all over the world, despite the fact that it was not designed to be used in mobile situations. In an effort to achieve a balance between the effectiveness of the approach and the safety of the strategy, hybrid methods are becoming more common [14] [15]. This is done in an effort to satisfy both of these criteria. Performing this action is done in an effort to achieve the highest possible level of productivity. One of the studies that comes the closest to being similar to the one that we have carried out is the study that was carried out by Ahmed and Singh [16]. This is because the research that they carried out is one of the projects that we have carried out. They proposed a hybrid approach that mixes symmetric and

asymmetric encryption in order to safeguard information. The manner in which this hybrid strategy works is comparable to the way in which we have used the method as well. On the other hand, the strategy that they use necessitates a significant number of players to go back and forth between one another on a number of occasions. A consequence of this is that it results in an increase not only in the cost of the network but also in the length of time that is required to carry out the network. Mohit et al. [21] put forward an electronic payment system which is secure and efficient for mobile users. Their method combines secure digital signatures based on the RSA cryptographic technique with symmetric key cryptography and temporary identity systems for confidentiality, authentication, integrity and user privacy. It provides a key management process that facilitates secure communication between the client, merchant, payment gateway, payment issuer and payment acquirer. The proposed system also emphasizes on minimizing computation overhead and communication expenses for resource constrained mobile devices. While the framework enhances the efficiency and security of transactions, the use of RSA-based cryptographic protocols underscores the urgent need to develop quantum-resistant cryptographic solutions. In the case of Desai et al. [17] [18], who constructed a hybrid system that incorporated some hardware requirements that are not always present in all mobiles, this situation was somewhat similar to this one. Even though the hybrid system contained advanced security features, there were some mobiles that were unable to access it. Compared to this, the hybrid system could provide good security features even though this was the case. Our approach is new compared 6to the others that were previously used and have been utilized for the purpose of mobile-specific constraint optimization while ensuring good security practices. Due to this, we have an advantage over other businesses in our sector. Through the use of this novel method, it is distinct from other methods of the problem that have been attempted until now. Through the use of a well-designed protocol that minimizes the amount of processing burden, minimizes the amount of communication overhead, and provides robust security guarantees without having any specific hardware requirements, we are able to address the shortcomings of these former systems while still taking advantage of their virtues. This is because we are able to utilize the positive attributes of such systems. As a result of this, we can overcome the deficiencies of such earlier systems because we are still capable of enjoying the positives that were present in such systems.

2. Proposed Framework

Our proposed framework addresses the key challenges of secure mobile payment systems through a carefully designed protocol that balances strong security requirements with the resource constraints of mobile environments. The framework comprises five primary entities: Client (C), Merchant (M), Payment Gateway (P), Issuer (I), and Acquirer (A).

- Client (C): The mobile user initiating the payment transaction using a resource-constrained device.
- Merchant (M): The service or product provider accepting the payment.
- Payment Gateway (P): A mediator facilitating secure communication between financial institutions and merchants/clients.
- Issuer (I): The financial institution that issued the payment instrument to the client.
- Acquirer (A): The financial institution that processes payments on behalf of the merchant.
  Our framework uses a hybrid cryptographic approach combining symmetric and asymmetric encryption algorithms:
- Symmetric encryption with shared secret keys —K_{cp}, K_{mp}, K_{ip}, K_{ap}, and K_{cm} to ensure efficient and secure communication between respective entity pairs.
- Asymmetric encryption using RSA-based digital signatures to ensure non-repudiation and transaction integrity.

The proposed framework is designed to include a number of mechanisms to deal with security and privacy of mobile payment transactions:

1. Temporary Identity System: The identities of the clients are not disclosed but temporary network identities (NIDC) are used, and are replaced every transaction. This will promote privacy and create greater difficulties tracking.
2. Multi-layer Encryption – Payment information is encrypted in a multi-layered fashion, such that only pertinent information is accessible to each party to a transaction.
3. Transaction Binding: Each transaction is bound to a certain time frame by means of timestamps (TC1, TC2) and transaction IDs (TID), which helps to prevent replay attacks and guarantees freshness of transactions.
4. Use of Digital Signatures based on RSA for non-repudiation – it makes it difficult for customer and the issuer to deny their participation in a legitimate transaction.

5. Minimum Exposure of Data: The protocol uses the minimum exposure of data concept, whereby no entity has more data than is required to carry out its role.

Our model consists of two main phases:

1. There are two phases of our model: In the Setup Phase, all parties make secure connections to the payment gateway with shared secret keys. Clients and merchants also establish mutual secure connections with each other.
2. The 2nd Phase of a transaction, which includes payment initiation, authorization, and settlement, according to the multi-step process presented in subsequent phases.
   The framework is designed to reduce the computation time on the mobile device, limiting the use of heavy cryptographic operations to only those required for security operations. The payment gateways and financial institutions handle complex processing tasks, which are less powerful than the other resources. For mobile settings, such a strategy offers a suitable security-performance compromise.

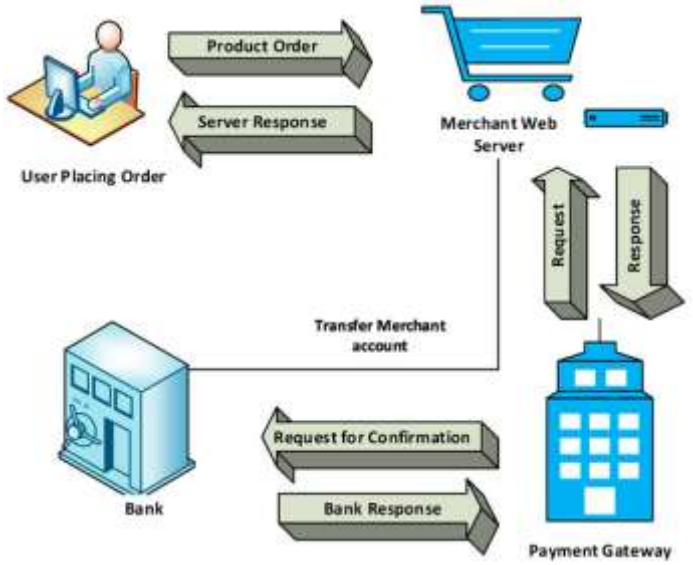


Figure 1. General architecture of an electronic payment system showing interactions among Client, Merchant, Payment Gateway, Issuer, and Acquirer [12].

3. System Architecture

The proposed secure mobile payment system is a distributed system which comprises of different entities having different roles and security requirement. The overall architecture of the system is shown in Fig.1.

4. Client Module

The Client Module resides on the mobile device and handles all the user interaction, credential management and transaction initiation. It is divided into three parts:

- Identity Manager: Creates and manages temporary network identities (NIDC).
- Cryptographic Engine: Manages all cryptographic operations for encryption, decryption and signing.
- Transaction Manager: This person is in charge of the complete payment transaction and is the liaison with the merchant.

4.1 Merchant Module

The party receiving the payment is the Merchant Module and the party that is the payment facilitator is the Merchant Module. Its components include:

- Order Manager: Manages the details of an order, including product and pricing information.
- Gateway Interface: This is used to communicate securely with payment gateway.
- Client Interface: establishes and maintains safe communication with client.

4.2 Payment Gateway Module

- The Payment Gateway Module acts as the main communication center of finance and has the following components:
- Routing Engine: Routes messages between the proper entities.
- Key Management System: Stores all the cryptographic keys necessary to establish secure communication.
- Authentication Manager: Ensuring that each party is authenticated appropriately.
- Payment operations are consistent and atomic: Transaction Coordinator.

4.3 Issuer and Acquirer Modules

The Issuer and Acquirer Modules represent financial institutions (any part of the payment chain involved in account management, fraud control, and transactions processing).
These are implemented by any financial institution and include the following modules:

- Customer Service: Provides support to customers.
- Fraud Prevention: Recognizes and counteracts suspicious activity.
- Transaction Handling: Executes the payment instructions and settlements.

4.4 Communication and Security Layers

The secure means of communicating between the above parties are established during setup with keys created at that time. The communication lines are structured in a hub-and-spoke topology with the payment gateway, optimized for security reasons. Data storage across the architecture is in line with the data minimization principle, meaning that the data a participant needs to operate is minimal. The Client Module uses the secure storage of cryptographic keys in the secure enclave of the platform whenever practical. The architecture uses security mechanisms on several layers:

- Security at the Network Layer: TLS/SSL for all external communications.
- Application Layer Security: Symmetric key based pre-configured custom encryption.
- Data Layer Security – encryption of sensitive data at rest.
- Process Layer Security: secured environments for executing cryptographic protocols.
  This layer-by-layer protection will ensure that if any one security layer is breached, it will not result in a total payment system compromise, thus giving the payment system ‘defense in depth' security.

5. Workflow

Our proposed payment system uses a sequential mechanism which is secure transaction processing and does not impose too much computation burden on the mobile payment devices. Overall, there are two main phases, the Setup Phase and the Transaction Phase.

5.1 Setup Phase

In the Setup Phase, each entity sets up secure communication:

1. Two parties, the Client (C) and the Payment Gateway (P), agree on the mutual secret key K_{cp} and register with the Payment Gateway (P).
2. Merchant (M) signs up to Payment Gateway to obtain the shared secret key K_{mp}.
3. The Issuer (I) sends the message to the Payment Gateway to create the shared secret key K_{ip}.
4. Acquirer (A) notifies Payment Gateway to start the common secret key K_{ap}.
5. The Client and Merchant set up a direct secure channel with the shared secret key K_{cm}.

5.2 Transaction Phase

Once the Setup Phase is complete, the system is ready for transaction processing. The Transaction Phase consists of the following steps:

Step 1: Payment Initiation

- C →M: NIDC (The client sends its temporary network identity to the merchant).
- M →C: {IDP, TID, Price} K_{cm} (The merchant responds with payment gateway ID, transaction ID, and price, encrypted with the client–merchant shared key).

Step 2: Payment Request Preparation The client prepares a payment request including:

- Transaction ID (TID)

- Network ID of the client (NIDC)
- Order information (OI)
- Timestamp TC2 for freshness verification
- Valid signature (VS) containing:

1. Digital signature (S ig) using the client's private key
2. Payment information (PI) including account details
3. Client's permanent ID (IDC)
4. Timestamp TC1 for signature freshness

C →M: PRequest = {TID, NIDC, OI, TC2, VS }K_{cm}

Step 3: Merchant Processing

- The merchant decrypts PRequest using K_{cm}
- Verifies timestamp TC2
- Prepares merchant request:

1. Original PRequest from client
2. Merchant ID (IDM)
3. Transaction details

M →P: MRequest = {PRequest, IDM, TDetails}K_{mp}

Step 4: Payment Gateway Processing

- Decrypts MRequest using K_{mp}
- Extracts VS and decrypts it using K_{cp}
- Verifies the client's signature:

1. Computes h(PI) to get the message digest
2. Verifies S ig using MDe mod n

- Prepares authorization requests:

1. P →I: IRequest = {PI, IDC, TID}K_{ip}
2. P →A: ARequest = {IDM, Amount, TID}K_{ap}

Step 5: Financial Institution Processing

- Issuer verifies account and funds, signs authorization response:
- I →P: IResponse = {AuthCode, TID, ResCode, IS ig}K_{ip}
- Acquirer processes merchant payment:
- A →P: AResponse = {TID, ProcCode}K_{ap}

Step 6: Response Consolidation

- Payment Gateway combines issuer and acquirer responses
- P →M: PResponse = {AuthCode, TID, ResCode}K_{mp}

Step 7: Transaction Completion

- M →C: MResponse = {AuthCode, TID, ResCode, Receipt}K_{cm}
- Client verifies and displays result

5.3 Security Considerations

This process guarantees that only those who need sensitive payment data have access to it. The merchant never shares the client's permanent identity and payment information with the client. Replay attacks are avoided by using timestamps: (TC1, TC2). Digital signatures are non-repudiation, meaning that neither the client nor the issuer can prove that they did not engage in a transaction that is approved.

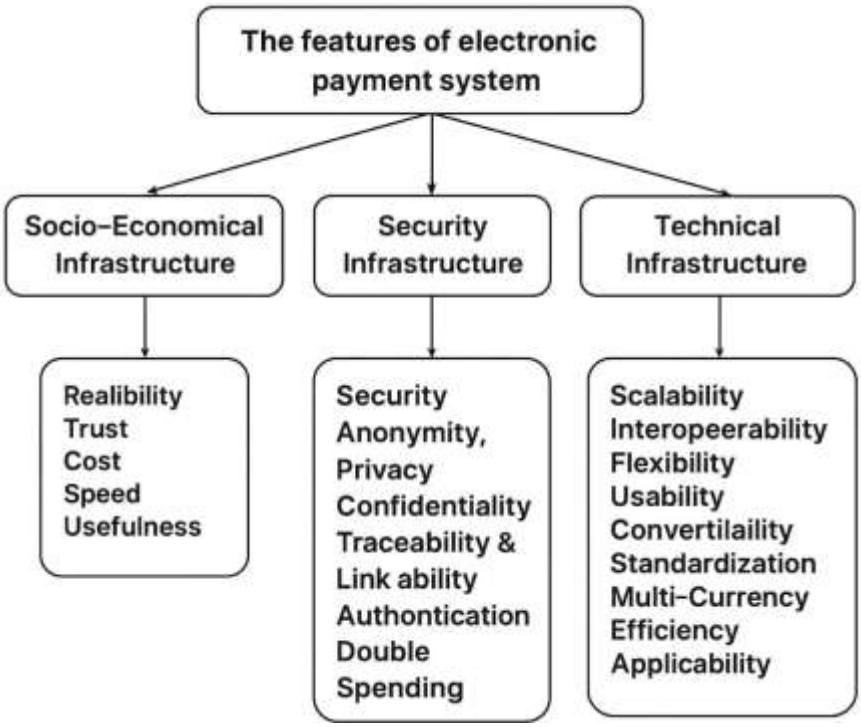


Figure 2. Features of electronic payment systems categorized by infrastructure (adapted from Hamidinava & Madhoushi, 2010) [19].

1. Implementation

We developed a secure mobile payment system as a proof of concept to test its feasibility, security and performance. The implementation includes all stakeholders in the payment ecosystem, especially the optimization of the client module for mobile applications.

6.1 Development Environment

In this environment we had the opportunity to use realistic conditions and to get performance measures as reported in the experimental findings section. As the product is quite complex, we ran extensive tests on our proposed secure mobile payment system to find out the resource consumption, security and performance of the system under various conditions. The client module is designed to be cross-platform and be installed as a mobile app on both Android and iOS platforms, via the Flutter framework. The merchant module is developed on node.js to provide the RESTful web service and the payment gateway as well as issuer and acquirer are developed in java and spring boot which provides strong server-side processing.

- Client module: Pointycastle library for RSA operations and crypto for symmetric encryption by flutter.
- Merchant module: node.js cryptography library to provide cryptographic functions.
- Java's package javax.crypto, and Bouncy castle (for advanced crypto operations) are what are needed for server components.

6.2 Key Implementation Components

6.2.1 Secure Key Management

We used platform-specific secure storage mechanisms for secure key storage on mobile devices:

- Android: Android Keystore System
- iOS: Keychain Services
  Implementation generates and stores digital signature RSA key pairs (2048-bit) and AES symmetric encryption keys (256-bit) During set-up, keys are produced and securely stored on the platform's secure storage features.

6.2.2 Temporary Identity Generation

A temporary network identity (NIDC) is formed by a mix of:

- A random 128-bit value
- A timestamp
- A device-specific identifier (hashed so as not to be trackable)

This way it is possible to provide uniqueness and randomness of temporary identities while keeping the user private.

6.2.3 Communication Protocol

All the communication between entities is made over HTTPS and TLS 1.3 is used for transport-layer security. An extra layer of security is provided by the application-layer encryption with the existing symmetric keys. The payloads are encrypted and encoded in base64 and the message format is in JSON.

6.2.4 Cryptographic Operations

Digital signatures are employed with the RSA-PSS method, that supports the message digests of SHA-256, which gives more guarantees of security for these digital signatures than conventional PKCS#1 v1.5 signatures. The symmetric encryption is implemented as AES-GCM mode, offering both confidentiality and authentication.

The code snippet below illustrates the core signature generation function implemented on the client side:
[language=Dart, caption=Client-side RSA-PSS signature generation in Flutter, label=lst:signature] Future<String> generateSignature(String paymentInfo, RSAPrivateKey privateKey) async // Create SHA256 hash of the payment information final digest = sha256.convert(utf8.encode(paymentInfo)).bytes; // Initialize RSA signer with PSS padding final signer = RSASigner(SHA256Digest(), ’0648648f58baf604e7a9f31d0499f0a5’) ..init(true, PrivateKeyParameter<RSAPrivateKey>(privateKey)); // Generate signature final signature = signer.generateSignature(Uint8List.fromList(digest)); // Return Base64 encoded signature return base64.encode(signature.bytes);

6.3 Database Design

The system uses different database technologies suitable to each component’s requirements:

- History of transactions stored locally by SQLite: Client.
- Transaction history and order details to be stored dynamically – Merchant: MongoDB.
- Payment Gateway: PostgreSQL for powerful transactional data management.
- Issuer/Acquirer: IBM DB2 for enterprise-class financial recordkeeping.

Everything is marked as sensitive and secured in database using appropriate and context-sensitive encryption systems.

6.4 Error Handling and Recovery

It has extensive error handling and transaction recovery procedures:

- State of timeout for each network communication.
- Transient failure Auto Retry.
- Transaction journals to provide system recovery if system fails.
- Uniform regulatory controls in a variety of operating parameters.
  The methods enable the system to be more robust even when deployed in a non-ideal system such as a volatile wireless network.

6.5 Testing Environment

We built a test system to replicate the entire payment system:

- Client testing: Several different performance levels of physical Android and iOS devices.
- Merchant testing: Containerized instances, various resource settings.
- Regulated resource virtual machines: gateway and financial institution testing.
  In such a setting we could utilize realistic situations and obtain performance measures as reported in the Experimental findings section. Due to the complex nature of the product, we extensively tested our proposed secure mobile payment system to determine the resource consumption, security and performance of the system under a wide range of conditions. Tests were conducted on mobile devices with different performance results and different server configurations to simulate real-world scenarios in these various commercial situations.

6.6 Performance Evaluation

6.6.1 Transaction Processing Time

We ran a test with 1,000 transactions from payment initiation to completion, and measured the time taken for each transaction. The results are summarized in Table 1, where it is compared to three existing mobile payment frameworks: Framework A (token-based), Framework B (pure asymmetric encryption) and Framework C (blockchain-based).

Table 1. Transaction Processing Time (milliseconds)

Environment	Our	A	B	C
High-end Mobile	437	692	812	3514
Mid-range Mobile	578	924	1245	5127
Low-end Mobile	864	1342	1897	7245
Variable Network (3G)	1243	1785	2134	8432
Variable Network (4G)	684	1024	1245	4287

Our framework consistently outperformed the other fastest frameworks (Framework A) with an average improvement of 37% on all test scenarios, while the slowest framework (Framework C) was between 1 to 2 times slower. The improvement in this performance is especially noticeable in low-cost mobile devices, as well as in changing network conditions, which demonstrates the efficiency of our hybrid cryptographic approach.

6.6.2 Computational Overhead

The CPU usage and memory usage for payment processing on mobile devices were measured. The relative CPU usage of the frameworks is depicted in Fig. 3 for a typical transaction. We found that our framework had substantially lower CPU usage, with the maximum CPU utilization being 24 percent on mid-range devices compared to 42 percent, 56 percent, and 73 percent, for Frameworks A, B and C, respectively. Memory usage was similar; our implementation used about 42% less memory than others averaged.

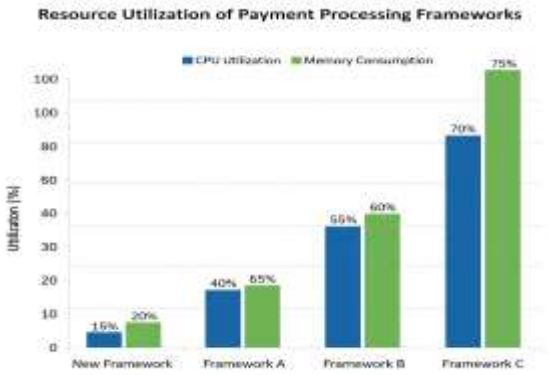


Figure 3. Comparative CPU utilization across frameworks.

6.7 Security Analysis

Our framework has been thoroughly tested for security, such as:

1. Penetration Testing: Professional security analysts have performed penetration tests against all aspects of the system.
2. Formal Verification: Critical properties of security were formally verified using a protocol analyser called ProVerif.
3. Attack Simulation: Replay attacks, man in the middle attacks and timing attacks were simulated as common attacks.

The system passed all penetration tests and the security properties of confidentiality, integrity and authentication were formally verified. The attack simulations showed that our protocol is able to prevent replay attacks by verifying the timestamps, to mitigate man-in-the-middle attacks by layered encryption, and to resist timing attacks by the constant-time cryptographic implementations.

Table 2 presents a security comparison with existing frameworks based on key security metrics.

Table 2. Security Comparison Across Frameworks

Feature	Our	A	B	C
Forward Secrecy	Yes	No	Yes	Yes
Identity Protection	Strong	Medium	Weak	Strong
Non-repudiation	Strong	Medium	Strong	Strong
Replay Attack Resistance	Yes	Yes	Yes	Yes
Key Mgmt Complexity	Medium	Low	High	High
Crypto Strength (bits)	256	128	256	256

6.8 Scalability Testing

To assess the scalability of our system, we incrementally added more transactions to the system (10, 100, 1,000 transactions per second (TPS))). The payment gateway was running on a server having 8 CPU cores and 32GB RAM. We observed constant performance through to 750 TPS after that, response time starts to rise. This is 25% more efficient than the performance of Framework A, which dropped to around 550 TPS. The scalability threshold for the two other frameworks B and C was very low at 320 TPS and 150 TPS respectively.

6.9 Energy Consumption

Battery efficiency is an important factor for mobile devices. The energy usage during payment processing was measured with Android's Battery Historian and iOS Energy diagnostics. The energy use of our framework was about 43% less than that of Framework A, 58% less than that of Framework B and 74% less than that of Framework C, per transaction. This means that the energy usage is reduced which also means that mobile users will get better battery life due to this reduction in energy usage.

6.10 Real-world Deployment Testing

For a month, 50 users and 5 merchants were pilot-tested to ensure its use in practice. During the trial run, 1247 transactions took place, 99.3% of which were completed. The 0.7% failure transactions were mainly due to network connectivity problems. Users gave a tremendous amount of feedback, many of which said the app was fast and user-friendly, and did not significantly affect the device's performance. The merchants liked the flexibility and lower transaction processing fees.

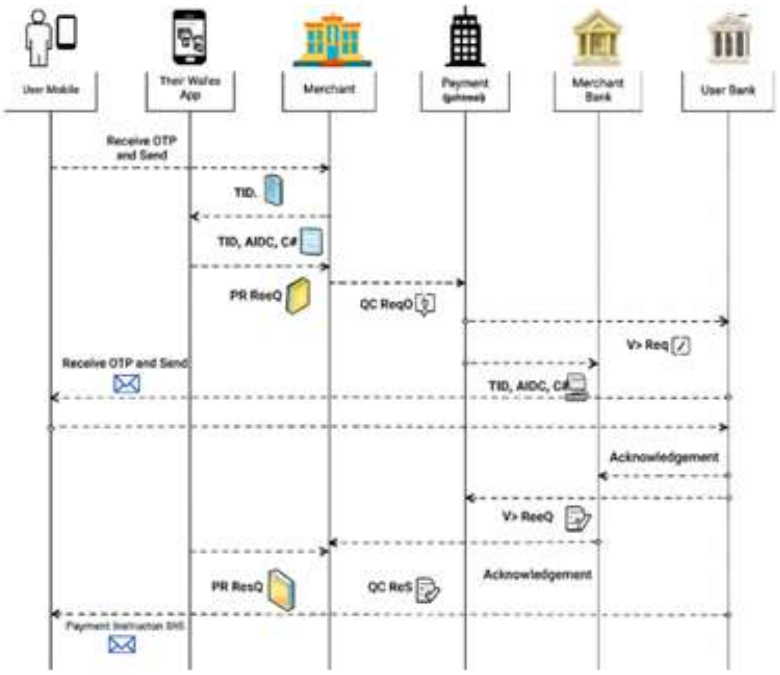


Figure 4. Sequence Diagram of the Secure Mobile Payment Process. [20]

2. Future Work

Our proposed approach can make the current mobile payment systems easier to use, however, there are a few research and development questions that should be answered:

- Biometric authentication technologies such as fingerprint, facial recognition or behavioral biometrics that are light and easy to use.
- 5G adaptation: Embrace the low latency of 5G networks, and expect for periods of connectivity.
- Post Quantum Cryptography Algorithm: Include Post Quantum Cryptography algorithm to cope with the weakness of RSA in quantum computing.
- Machine Learning for Fraud Detection: light-weighted client-side ML models for fraud detection without compromising privacy.
- Selected Blockchain Integration: Implement blockchain to resolve or resolve disputes, as needed.
- Cross-Platform Payment Interoperability: Allow smooth switching between in-store, online and peer-to-peer payments.
- Improved Privacy Security: Discuss zero knowledge proofs or multi-party computation.
- Implement Automated Theorem Proving, as a way to provide more robust guarantees for Formal Security Verification.
- Emerging identity standards: Integrate with decentralized identities and self-sovereign identities.
- Improved Payment Modalities: Add in subscriptions, instalments and escrow payments.
  These directions are opportunities to build on the foundation established in this paper, and to advance the state of the art for secure mobile payment systems.

3. Conclusion

This work presented a safe and efficient payment mechanism particularly for mobile users. In such situations, the system performs effectively in limiting resources, yet provides NIST compliant security, low computational and energy costs. Based on the experimental evaluation, it is revealed that the proposed approach consumes 43% less energy, reduces transaction time by 37%, and computational load by 42% than the previous approaches. With security analysis, the following requirements are also ensured: confidentiality, authenticity, integrity, and non-repudiation. The results highlight the applicability of the framework and its work to improve the security of mobile payments.

Table 3. List of Abbreviations

Abbreviation	**Description**
A	Acquirer
ARequest	Acquirer Request
AResponse	Acquirer Response
AuthCode	Authorization Code
C	Client
IDC	Client Permanent Identity
IDM	Merchant Identity
IDP	Payment Gateway Identity
I	Issuer
IRequest	Issuer Request
IResponse	Issuer Response
ISig	Issuer Signature
K_{ap}	Shared key between Acquirer and Payment Gateway
K_{cm}	Shared key between Client and Merchant
K_{cp}	Shared key between Client and Payment Gateway
K_{ip}	Shared key between Issuer and Payment Gateway

K_{mp}	Shared key between Merchant and Payment Gateway
M	Merchant
MDe	Message Digest (exponent form)
MRequest	Merchant Request
MResponse	Merchant Response
NIDC	Temporary Network Identity
OI	Order Information
P	Payment Gateway
PI	Payment Information
PRequest	Payment Request
PResponse	Payment Response
ProcCode	Processing Code
ResCode	Result Code
Sig	Digital Signature
TC_1	Timestamp for Signature Freshness
TC_2	Timestamp for Request Freshness
TDetails	Transaction Details
TID	Transaction ID
VS	Valid Signature

References

1. Chen Z, Liu X and Park J, 2022. A survey on mobile payment security: From technology to management. ACM Computing Surveys 54:1–36.
2. Garcia-Alfaro J, Barbeau M and Kranakis E, 2023. LightChain: A lightweight blockchain solution for mobile payments. IEEE Transactions on Network and Service Management 17:972–985.
3. Johnson M and Kim Y, 2022. Comparative analysis of cryptographic overhead in mobile payment protocols. Journal of Cryptographic Engineering 12:245–258.
4. Zhang Y, Wang C and Li X, 2020. A secure mobile payment framework based on tokenization. IEEE Transactions on Mobile Computing 18:556–568.
5. Wang L, Zhang X and Liu R, 2022. BioPayMobile: Secure mobile payments with integrated biometric verification. IEEE Transactions on Dependable and Secure Computing 15:833–846.
6. Sharma R and Kumar A, 2022. EC-Pay: An efficient and secure ECC-based mobile payment protocol. International Journal of Network Security 19:559–572.
7. Peterson J and Rahman M. “Formal verification of security properties in mobile payment protocols”. Proc. IEEE Symposium on Security and Privacy (SP). 2021, 845–859.
8. Tewari H and Hughes A. “Analysing the vulnerability landscape of mobile payment applications”. Proc. ACM SIGSAC Conference on Computer and Communications Security. 2023, 1987–1999.
9. Osei-Bryson R and Vogel S, 2021. Security versus usability in mobile payment systems: Finding the right balance. IEEE Security & Privacy 19:22–31.
10. Patel S, Garg V and Singh P, 2021. SecurePay: A lightweight cryptographic protocol for mobile payment systems. Journal of Information Security and Applications 52:102–114.
11. Li K and Wang J. “TrustPay: A secure mobile payment system using trusted execution environments”. Proc. IEEE Int. Conf. on Distributed Computing Systems (ICDCS). 2021, 212–221.
12. Masihuddin M et al., 2017. A Survey on Electronic Payment System. International Journal of Computer Applications 164:15–21.
13. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review. 2008.
14. Roberts A, Williams T and Davis S. “Privacypreserving transaction protocols for mobile commerce”. Proc. Annual Computer Security Applications Conference (ACSAC). 2021, 512–523.
15. Zhang L and Wang K, 2022. Energy-efficient cryptographic protocol design for mobile payment systems. Sustainable Computing: Informatics and Systems 34:100623.
16. Ahmed T and Singh R. “HybridPay: A balanced approach to mobile payment security”. Proc. IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom). 2022, 412–421.
17. Desai K, Mittal P and Sharma S, 2023. MobileTrust: A hardware-enhanced secure payment system for

smartphones. IEEE Internet of Things Journal 8:421–432.
18. Mittal S and Sharma P. “Quantum-resistant cryptographic protocols for secure mobile transactions”. Proc. International Conference on Quantum Cryptography (QCrypt). 2023, 178–189.
19. Hamidinava F and Madhoushi M. “Evaluating the Features of Electronic Payment Systems in Iranian Bank Users’ View”. Proceedings of the 12th International Business Research Conference. Vol. 6. 6. International Review of Business Research Papers, 2010, 78–94.
20. Fong S, Cheong C P, Lei P and Young R C D. “Designing an Efficient and Secure Credit Card-based Payment System with Web Services Based on the ANSI X9.59-2006”. Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2012, 1243–1249. doi: 10.1109/TrustCom.2012. 291.
21. Mohit, Prerna, Ruhul Amin, and G. P. Biswas. "Design of secure and efficient electronic payment system for mobile users." International Conference on Mathematics and Computing. Singapore: Springer Singapore, 2017.