Simplified Hybrid Secure Algorithm for Mobile Banking Application

Roayat Ismail Abdelfattah, Sherif Awad and Mohamed E Nasr
Department of Electronics and Electrical Communications, Faculty of Engineering
University Tanta, Tanta 31511, Egypt
E-mail: royat_esmaeel@f-eng.tanta.edu.eg
E-mail: eng.sherifawad@gmail.com
E-mail: mohamed.nasr@f-eng.tanta.edu.eg

Abstract. M-banking is one of the main divisions of m-commerce. It provides daily banking operations to the customer using his mobile with the supported application. But the current applications are facing security challenges due to the limitations of mobile resources. There are two major schemes. First, the public key infrastructure (PKI) but, suffers the scalability and certificate management problems. Second, the identity-based public key cryptography (IB-PKC) but has key escrow problem. In this paper a modified algorithm to solve these problems by combining elliptic curve Signcryption with certificateless cryptography and enhance the security with lower computational time. From the results, the proposed algorithm has superior performance compared with earlier schemes. Also, it supports multiple types of messages (such as document and multimedia).

1. Introduction

Mobile banking is a way for a bank’s customer to access banking services using his cell phone to provide functions like balance query, debit application, payment transaction, electronic bill payments, and, etc. It reduces time and effort costs at the different bank branches. Understanding the mobile banking and payments market and ecosystem is critical in addressing the security challenges. The security in M-banking applications is important both at the administrative level and from the user perspective. So, the Cryptography is used to prevent the third parties from reading private messages. Usually, banks enable customers to access their banking accounts through one of these technologies the interactive voice response (IVR), the short messaging service (SMS) or the web-based mobile applications [1]. Cryptography techniques are used for secure communications by constructing protocols that prevent third parties from reading the private messages.

In a traditional public key cryptography (PKC), to be able to communicate with others, a public key should be associated with the owner by a certificate which is a signature issued by the trusted Certificate Authority (CA). That technique brings a large amount of computation, communication cost and certificate management problems. To solve those problems, identity-based cryptography is used. By using information related to the user identity, publicly known and unique in the whole system, as his public key. The advantage of identity-based cryptography is that only user’s identity is required to communicate. However, the user must completely trust KGC, which introduce as key escrow problem [2], [3].
2. Literature Review

Most of the existing security framework proposals using PKI but due to its associated problems a various software implementation is used to overcome resource limitation issues. This is the case of [4], which use RSA and PKI but the scheme has a large overheads in terms of memory, processing, and SMS usage both for digital signatures and public key validation. Another workaround for PKI that sacrifice usability is given in [5], which proposes users register themselves on a website before they start exchanging secure SMS. Another common strategy is the adoption of an auxiliary device with better computational capabilities, like a SIM card empowered by a co-processor and extra storage space for private keys [6], [7]. A few researchers introduced solutions to the mobile banking security problems using the identity-based cryptography. For example, Zhao and Aggarwal [8].

The Certificateless cryptography remove the need for certificates also solve the key escrow problem in ID-PKC. However, certificateless cryptography schemes have not appeared in the literature in association with applications. In Certificateless cryptography the KGC server is involved only in generating a user’s partial key without knowing the value of user’s secret key [2], [9] but they depend on the bilinear map which needs costly operations so it’s inefficient. This includes the uses of certificateless encryption [10], certificateless signatures and certificateless signcryption [11]. Recently, in [12] the authors introduced a secure mobile banking scheme based on certificateless cryptography, but it depends on the bilinear map which needs costly operations so it’s inefficient.

The proposed scheme applies the Hypred technique on mobile banking transactions without using the bilinear map or limited by text messages like in [2]. The proposed scheme enhances the security and reduces the computational time compared to the existing Hypred techniques like wireless networks [13] or IOT [2], [14] or Instant Message [3].

3. Cryptography Basic Assumptions

In this section, a brief introduction for several computational assumptions relevant to our proposal.

3.1. Elliptic curve cryptography

Let $F_p$ denotes the finite field of modulo of a large prime $p$. An elliptic curve $E$ over the finite field $F_p$ is defined via $Y^2 = X^3 + aX + b \mod p$ Where $a$ and $b$ satisfy $4a^3 + 27b^2 \mod p \neq 0$. A base point $P$ of the elliptic curve $E$ with order $n$ should satisfy $N*P=O$, where $N$ is the order of the base point $P$ and $O$ is a point at infinity of $E$. The points of elliptic curve $E_p (a, b)$ together with a point $O$ at infinity from an addition cyclic group $G_p$ with order $p$.

3.1.1. Elliptic Curve Discrete Logarithm Problem (ECDL). Finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible.

3.2. Certificateless Public Key Cryptography

The certificateless public key cryptography system, uses a key generating centre (KGC) to generate a partial private key and delivers it securely to the correct entities. The entity uses the partial private key with a secret key (not known to the KGC) to generate a full private key.

4. The Proposed Scheme Description

4.1. Security Requirements for the scheme

The proposed scheme should have the following security features.

- Confidentiality.
  The text should be set to single line the encrypted message must be correctly decrypted only by the specified recipient.
- Integrity.
  The recipient should be able to know that the message has not been tampered with.
• Authentication.
  The recipient should be able to verify the received message sender.
• Unforgeability.
  The scheme must be resistance to Type I attack which is the public key substitution attack and Type II attack which is the malicious KGC attack.
• Public Verifiability.
  A judge can authenticate that the signcrypted text is valid or not, without using the private keys.
• Non-Repudiation.
  The sender cannot deny sending the received message.
• Forward Secrecy.
  The secrecy of the messages is ensured, even if the sender's private key is compromised.
• Denial of Replay Attack.
  The recipient must be able to eliminate the duplicate message.

4.2. Certificateless Cryptography
Certificateless cryptography has three phases as shown in Figure 2 (Setup phase, Authentication phase and partial private key generation phase). The phases operations will be discussed in details.

![Figure 1](image1.png)  ![Figure 2](image2.png)

**Figure 1.** Client registration for mobile banking services.  **Figure 2.** Certificateless Cryptography

**Table 1.** Notations

| Notation | Description |
|----------|-------------|
| $q$      | Large prime number (160 bit) |
| $p$      | A base point on the elliptic curve |
| $n$      | The order of the base point |
| $H()$    | A collision-free one-way hash function |
| $\oplus$| The exclusive-OR operation |
| $\bot$  | The concatenation operation |
| $ACC$    | The Time Stamp |
| $ID_c$   | Client's account number |
| $X_K$    | The identity of a client |
| $Y_K$    | The x-axis value of the point $K_1$ |

4.2.1. Setup.
• KGC Server select an elliptic curve with the following parameters:
  $q$ prime number $\geq 2^{160}$
\[ E_y : Y^2 = X^3 + aX + b \ (Mod \ q) \]
\[ a, b, x, y \in E(F_p) \rightarrow 4x^3 + 27b^2 \ (Mod \ q) \neq 0 \]
\[ p = (P_x, P_y) \in E(F_p) \rightarrow \text{base-point of order (n)} \rightarrow n. \ p=O \]

- KGC Server Selects a random number as his master private key \(d_{KGC}\).
- Calculates his public key \(P_{KGC} = P \ast d_{KGC}\)
- Select a cryptography hash functions that satisfy \(H : \{0,1\}^* \rightarrow Z_q^*\)
- KGC Server Publishes \(\{E, p, n, H, P_{KGC}\}\) as the public parameters.

4.2.2. Client Registration phase.
- The client must apply for the service by meeting the bank’s employee Figure 1 to check his Identity and his account, then the client provides: mobile station international subscriber directory number (MSISDN), which will be used as his ID (ID_c).
- Approving procedures: The employee opens offline connections with:
  - Bank’s database server: check if there exists such client with such account number.
  - KGC server: insert the information then, generates a username, password.
- The client receives a username, password.

4.2.3. KGC Server Authentication phase.
- The client encrypts his username (hashed) by his password, then send it to KGC Server with the generated partial public key and the timestamp.
- KGC server receives the client’s encrypted message:
  - Decrypts it.
  - Check the received ID has the same hashed user name.
  - Encrypt the generated partial private key by client’s password.
  - Send back to the client.
- The client receives the encrypted message that received from the KGC server:
  - Decrypts it.
  - Client accepts the KGC server, otherwise rejects the connection.

4.2.4. Partial Private key Generation Phase.
- A client selects a random number as his secret key \(d_{c1}\).
- Calculates the partial Public Key \(P_{c1} = d_{c1} \ast P\)
- Send \(\{T, ID_c, P_{c1}\}\) to the KGC Server.
- KGC server chooses random number \(r_{cKGC} < n\), \(R_{cKGC} = r_{cKGC} \ast P\)
- KGC server computes client partial private key then delivers it to the client.
  - \(d_{c2} = [r_{cKGC} \ast d_{KGC} \ast H(ID_c \mid P_{KGC} \mid R_{cKGC} \mid P_{c1})] \ Mod n\)
- The client Verifies the received partial private key using \(R_{cKGC}\) and the public parameters.
  - if \(P_{c2} = R_{cKGC} \ast P_{KGC} \ast H(ID_c \mid P_{KGC} \mid R_{cKGC} \mid P_{c1})\), then accept \(d_{c2}\)
- A Client full keys:
  - Full secret key \(d_c = (d_{c1}, d_{c2})\)
  - Full Public key \(P_c = (P_{c1}, R_{cKGC})\)
- A Bank full keys:
  - Full secret key \(d_b = (d_{b1}, d_{b2})\)
  - Full Public key \(P_b = (P_{b1}, R_{bKGC})\)
The client computes \( P_{b2} = Rb_{KGC} + P_{KGC} \ast H(ID_b || P_{KGC} || Rb_{KGC} || P_{b1}) \) then stores it.

4.3. Elliptic curve Signcryption Cryptography
A client checks for changes in public parameters before performing the Signcryption process, if there is a change the client must repeat the KGC server authentication phase and the partial key generation phase. The Signcryption and Un-Signcryption process as follows.

4.3.1. Signcryption
- The sender chooses a random value \( r_e < n \).
- Computes the session key.
  - Computes \( z_e = P \ast r_e \)
  - Computes \( K_1 = r_e \ast P_{b2} = (X_{K_1}, Y_{K_1}) \)
- Encrypts messages.
  - Cipher text \( C = X_{K_1} \oplus M \)
  - New ACC \( Acc' = (T, Acc, ID) \oplus y_{K_1} \).
- Generates S, R.
  - \( h = H(T \| Acc' \| C \| ID_e \| z_e \| P_{e1} \| P_{b1} \| P_{KGC}) \)
  - \( S = (r_e - h \ast [d_{c1} + d_{c2}]) \mod n \)
  - \( R = h \ast [P_{c1} + P_{c2}] \)
- Sends the generated values \{C, R, S, ACC'\}.

4.3.2. Un-Signcryption
- Receives the values \{C, R, S, ACC'\}.
- Computes the session keys.
  - Computes \( z_e = P \ast S + R \)
  - Computes \( K_1 = z_e \ast d_{b2} = (X_{K_1}, Y_{K_1}) \)
- Get the original values \( (T, Acc, ID) = Acc' \oplus y_{K_1} \).
- Checks Acc if exists and the timestamp is larger than the stored then, continue.
- Validates the cipher text.
  - \( h = H(Acc' \| C \| ID_e \| z_e \| P_{e1} \| P_{b1} \| P_{KGC}) \)
  - \( P_{c2} = Rc_{KGC} + P_{KGC} \ast H(ID_e \| P_{KGC} \| Rc_{KGC} \| P_{e1}) \)
  - Accept the cipher if and only if \( R = h \ast [P_{c1} + P_{c2}] \).
- Retrieve the message \( M = X_{K_1} \oplus C \).

5. The Proposed Scheme Analyses

5.1. Proof of key retrieval.
\[
\begin{align*}
K_1 &= z_e \ast d_{b2} = r_e \ast p \ast d_{b2} \\
K_1 \oplus p \ast (r_{b_{KGC}} \ast d_{KGC} \ast H(ID_b \| P_{KGC} \| Rb_{KGC} \| P_{b1})) &= z_e + Rb_{KGC} + P_{KGC} \ast H(ID_b \| P_{KGC} \| Rb_{KGC} \| P_{b1}) = r_e \ast P_{b2} = K_1
\end{align*}
\]
5.2. Encryption / decryption algorithm.

The proposed scheme uses AES 128-bit GCM mode to encrypt the message between the Client and KGC server, it provides both authentication and encryption. The key is the Client password with applying the Key derivation function (KDF) with unique initialization vector for every encryption.

5.3. Security analysis

**Theorem 1**: The key agreement process is secure under the ECDLP\(^1\) assumption.

**Proof**: Assume that KGC has generated private keys for a client C and the bank B, and the adversary is denoted as A. We can prove the security as follows:

a) client C chooses a number \(r_c\) randomly and calculates
\[
z_c = p \cdot r_c \quad \text{and} \quad K_1 = r_c \cdot P_{b2}
\]
Then, obtain the current timestamp T and calculates S, R
client’s private key \(d_{e1} + d_{e2}\) and send the organized message to the bank B.

b) When the bank B has received the message, he will retrieve the session key.
\[
\Rightarrow \begin{align*}
z_c &= p \cdot S + R \\
K_1 &= z_c \cdot d_{b2}
\end{align*}
\] (1)

c) The Bank B will verify T, ACC and R
\[
\Rightarrow \begin{align*}
h &= H(\text{Acc'} \ || \ ID_c) = \text{Acc'} \oplus y_K, \\
P_{c2} &= R_{c_{KGC}} + P_{KGC} \cdot H(ID_c \ || \ P_{KGC} \ || \ R_{c_{KGC}} \ || \ P_{c1}) \\
R &= h \cdot [P_{c1} + P_{c2}]
\end{align*}
\] (2)

Adversary A can only obtain S, R, \(P_{c2}\) and the public parameters. In this case, obtaining \(K_1\) without \(d_{e1}, d_{e2}\) and \(d_{b2}\) should solve the ECDL problem. In addition, client C and bank B can detect forgery and replay attacks by verify if received
\[
R = h \cdot [P_{c1} + P_{c2}] \quad \text{and} \quad T.
\]

**Theorem 2**: The scheme resistance to Type I attack (the public key substitution attack).

**Proof**: Adversary A has access to the system public parameters but not client secret key

a) Adversary A chooses a random number \(d_{A1}\) gets client partial private key \(d_{c2}\) so the full private key \((d_{A1}, d_{c2})\) and replace the public keys to be \((P_{A1}, RC_{KGC})\) instead of \((P_{c1}, RC_{KGC})\).

b) Adversary A generates the K1, then calculates
\[
\Rightarrow \begin{align*}
h &= H(\text{Acc'} \ || \ ID_c \ || \ z_c) = \text{Acc'} \oplus y_K, \\
S &= (r_c - h \cdot [d_{A1} + d_{c2}]) \ Mod \ n \\
R &= h \cdot [P_{A1} + P_{c2}]
\end{align*}
\] (3)

c) The Bank B retrieve they key then gets T, ACC then verify them and calculates
\[
\Rightarrow \begin{align*}
P_{c2} &= R_{c_{KGC}} + P_{KGC} \cdot H(ID_c \ || \ P_{KGC} \ || \ R_{c_{KGC}} \ || \ P_{c1})
\end{align*}
\] (4)
Finally, it’s clear that received \(R \neq h \cdot [P_{c1} + P_{c2}]\). Hence, the type I attack is unsuccessful.

**Theorem 3**: The scheme resistance to Type II attack (malicious KGC attack).

**Proof**: Adversary A has access to KGC master key and can’t replace the client C partial private key.

---

\(^1\) Elliptic curve discrete logarithm problem (ECDLP).
Adv. A can’t get the value of the client secret key \( d_{c1} \), he needs to solve ECDLP which is by definition computationally infeasible.

b) Adversary A forges the full private key to be \( (d_{c1} , d_{A2}) \) and the public key is \( (P_{c1} , RA_{KGC}) \) then using the same equations:

\[
\begin{align*}
S &= (r_c - h \ast [d_{c1} + d_{A2}]) \mod n \\
R &= h \ast [P_{c1} + P_{A2}]
\end{align*}
\]

Finally, it’s clear that received \( R \neq h \ast [P_{c1} + P_{A2}] \). Hence, the type II attack is unsuccessful.

**Theorem 4:** The session key is secure from the attacker even if the private key of the Client is being compromised (Forward Secrecy).

**Proof:** Adversary A has access to client C private key.

a) Adversary A can only obtain \( S, R, P_{c2}, d_{c1}, d_{A2} \) and the public parameters.

b) Try to calculate the session key \( K_1 = r_c \ast P_{h2} \) so, he needs to calculate \( r_c \).

c) Try to get \( r_c \) from \( S = (r_c - h \ast [d_{c1} + d_{A2}]) \mod n \) but A needs to calculates \( h \).

Finally, it’s clear that to get \( h \), he needs \( r_c \) which is the unknown value he needs form the start. Hence, the forward secrecy property of the scheme is verified.

**Theorem 5:** A judge can verify if the specified client initiated the message or not without the needs to client’s private key (public verifiability).

**Proof:** Judge J has access to only \( (ID_c, C, S, P_{c1}, P_{b1}, Acc', P_{KGC}, R_{c1}) \), all values are public then, he performs the following:

a) Compute \( (z_c) = p \ast S + R \)

b) Compute \( h = H(\text{Acc}' || C || ID_c || z_c || P_{c1} || P_{b1} || P_{KGC}) \)

c) \( P_{c2} = R_{c1} + P_{KGC} \ast H(ID_c || P_{KGC} || R_{c1} || P_{c2}) \)

Finally, if \( R = h \ast [P_{c1} + P_{c2}] \) so, the specified client initiated the message.

### 6. Results and Dissections

#### 6.1. Computational Cost

Let Hash, Expo, ECPA, ECPM, and Pairing denote the execution time of the following: one hash function operation, one modular exponentiation operation, one elliptic curve point addition, one-point (scalar) multiplication, and one bilinear map, respectively. Using the experimental results obtained in [9], the hash time is the time unit: Expo = 600 Hash, ECPA =13 Hash, ECPM = 72.5 Hash, Pairing = 1550 Hash. Based on Hash = 0.014 ms.

| Table 2. Number of operations. |
|---|---|---|---|---|
| Ref | Exp | Pair | ECPM | ECPA | H () |
| [13] | 0 | 0 | 15 | 5 | 13 |
| [2] | 0 | 0 | 11 | 3 | 8 |
| [14] | 2 | 2 | 13 | 7 | 14 |
| [3] | 0 | 0 | 16 | 4 | 7 |
| [12] | 0 | 2 | 10 | 2 | 10 |
| Proposed | 0 | 0 | 15 | 6 | 7 |
Table 3. Security Services.

| Ref  | Confidentiality | Integrity | Authentication | Unforgeability | Public verifiability | forward secrecy | Resistance to Replay attack | Resistance to type II attack | Key escrow resistance | Resistance to type I attack | Multimedia & document message |
|------|-----------------|-----------|----------------|----------------|----------------------|-----------------|----------------------------|---------------------------|------------------------|---------------------------|-------------------------------|
| [13] | Yes             | Yes       | Yes            | Yes            | No                   | Yes             | Yes                        | Yes                       | Yes                    | Yes                       | Not mentioned               |
| [2]  | Yes             | Yes       | Yes            | Yes            | Yes                  | Yes             | No                         | No                        | Yes                    | No                        | Not mentioned               |
| [14] | Yes             | Yes       | Yes            | Yes            | No                   | Yes             | Yes                        | No                        | Yes                    | No                        | Not mentioned               |
| [3]  | Yes             | Yes       | Yes            | Yes            | Yes                  | Yes             | No                         | No                        | Yes                    | No                        | Yes                           |
| [12] | Yes             | Yes       | Yes            | Yes            | No                   | No              | Yes                        | No                        | Yes                    | No                        | No                             |
| Proposed | Yes             | Yes       | Yes            | Yes            | Yes                  | Yes             | Yes                        | Yes                       | Yes                    | Yes                       | Yes                           |

In [3] they used only the KGC Server to generate the private key for all entities using it’s master key and the user Identity which is unique in the whole system. However, the user must completely trust KGC, which can impersonate any user to sign or decrypt any message. This issue is known as key escrow problem [15]. But the proposed paper overcomes it by allowing the KGC server to generate only the partial private key not the full private key. Also, the proposed scheme has less computational time Figure 3.

The comparison shows that the proposed scheme achieves a less computational cost compared with Hassouna [12] and Li [14] Figure 3. The proposed scheme has more computational time than Omala [13], but offers more essential security features like public verifiability Table 3.

Our scheme offers the resistance to key escrow problem compared to Nguyen [2], furthermore other features shown in Table 3.

7. Conclusion
The paper discusses the advantage of the proposed Simplified Hybrid Secure Algorithm for Mobile Banking Application, the benefits of using Elliptic curve certificateless (CL) cryptography which overcomes many problems presented in the other schemes like PKI scheme and ID-PKC and enhancing the security with low computational time. Also, the flexibility to be used in mobile banking
applications and the ability for sending Multimedia and documents message has been achieved in the proposed scheme.

8. References

[1] M. O. Albasheer and E. B. M. Bashier, “Enhanced model for PKI certificate validation in the mobile banking,” in 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEE), 2013, pp. 470–476.

[2] K. T. Nguyen, N. Oualha, and M. Laurent, “Lightweight certificateless and provably-secure signcryption system for the internet of things,” in Trustcom/BigDataSE/ISPA, 2015 IEEE, 2015, vol. 1, pp. 467–474.

[3] Z. Wang, Z. Ma, S. Luo, and H. Gao, “Enhanced Instant Message Security and Privacy Protection Scheme for Mobile Social Network Systems,” IEEE Access, vol. 6, pp. 13706–13715, 2018.

[4] J. P. Albuja and E. V. Carrera, “Trusted SMS communication on mobile devices,” in 11th Brazilian Workshop on Real-Time and Embedded Systems, Pernambuco, Brazil, 2009, pp. 165–170.

[5] A. De Santis, A. Castiglione, G. Cattaneo, M. Cembalo, F. Petagna, and U. F. Petrillo, “An extensible framework for efficient secure SMS,” in Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, 2010, pp. 843–850.

[6] G. C. Pereira et al., “SMSCrypto: A lightweight cryptographic framework for secure SMS transmission,” Journal of Systems and Software, vol. 86, no. 3, pp. 698–706, 2013.

[7] H. Rongyu, Z. Guolei, C. Chaowen, X. Hui, Q. Xi, and Q. Zheng, “A PK-SIM card based end-to-end security framework for SMS,” Computer Standards & Interfaces, vol. 31, no. 4, pp. 629–641, 2009.

[8] S. Zhao, A. Aggarwal, and S. Liu, “Building secure user-to-user messaging in mobile telecommunication networks,” in Wireless Telecommunications Symposium, 2008. WTS 2008, 2008, pp. 151–157.

[9] M. H. Ibrahim, S. Kumari, A. K. Das, and V. Odelu, “Jamming resistant non-interactive anonymous and unlinkable authentication scheme for mobile satellite networks,” Security and Communication Networks, vol. 9, no. 18, pp. 5563–5580, 2016.

[10] S. S. Vivek, S. S. D. Selvi, and C. P. Rangan, “CCA secure certificateless encryption schemes based on RSA,” in Security and Cryptography (SECRIPT), 2011 Proceedings of the International Conference on, 2011, pp. 208–217.

[11] W. Xie and Z. Zhang, “Efficient and provably secure certificateless signcryption from bilinear maps,” in Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on, 2010, pp. 558–562.

[12] M. Hassouna, N. Mohamed, and E. Bashier, “A Secure Mobile Banking Scheme Based on Certificateless Cryptography in the Standard Security Model,” International Journal of Computer Applications, vol. 74, no. 9, pp. 1–6, 2013.

[13] A. A. Omala, N. Robert, and F. Li, “A provably-secure transmission scheme for wireless body area networks,” Journal of medical systems, vol. 40, no. 11, p. 247, 2016.

[14] F. Li, Y. Han, and C. Jin, “Certificateless online/offline signcryption for the Internet of Things,” Wireless Networks, vol. 23, no. 1, pp. 145–158, 2017.

[15] G. Zhang and X. Wang, “Certificateless encryption scheme secure in standard model,” Tsinghua Science & Technology, vol. 14, no. 4, pp. 452–459, 2009.