Framework for Secure Mobile Banking Application Using Elliptic Curve Cryptography & Image Steganography

J. Saranya¹, S. Thirumal²

¹M.Phil research Scholar A.A.A. Govt. Arts College, Cheyyar, T.V. Malai Dt, India
²Assistant Professor & Head, of Computer Science, A.A.A govt. Arts College, Cheyyar, T.V. Malai Dt, India

Abstract: The Wide-expansion of mobile telecommunication technology mobile banking emerged as a new type of financial services and can provide efficient and effective financial services for clients. Mobile banking is a way for the customer to perform banking actions on his or her cell phone or other mobile device. It is a quite popular method of banking that fits in well with a busy, technologically oriented lifestyle. Framework conditions for mobile banking services differ from country to country but one thing is certain: the future of mobile banking depends on getting the security right. In this paper, we present a new way of securing mobile banking. We introduce a system which makes use of Elliptic curve cryptography and RGB Intensity Based Randomized pixels with variable Bits image Steganography [5]. Elliptic Curve Cryptography suites well for resources constraint devices like mobile phones and PDA, because of its less computation time, short key’s length, fast digital signature, flexibility and less resource consumption.

Keywords: Mobile banking, Elliptic curve cryptography, Steganography, Trusted server

1. Introduction

Mobile Banking scores over Internet Banking because it enables ‘Anywhere Anytime Banking’ and also it reduces the customer requirement to just a mobile phone. The biggest advantage Mobile Banking provides to the banks is that it helps to cut down the costs as it’s even more economic than providing tele-banking facilities where banks have to keep hundreds of tele-callers. Additionally, Mobile Banking helps banks to upgrade the quality of services and nature of customer relationship management. But Security concerns are the single biggest factor inhibiting consumer acceptance of mobile banking. The rest of the paper is organized as follows. Section 2 discusses our proposed system. Section 3 represents results of the proposed system. Finally, section 4 discusses the conclusion and future work.

2. Related Works

The most popular type of mobile banking is Short Message Service (SMS). The advantages of using SMS are that it is relatively inexpensive and it is reliable for sending non-sensitive messages. SMS messages are sent asynchronously. When a message is submitted for sending, the service provider will keep the sending message in its message buffer until the message is delivered to the destination mobile phone. The problems with SMS banking are that the SMS message is not entirely secured. There are many flaws in the GSM architecture which lead to security shortfalls for SMS banking. In USSD mobile banking the verification depends only on the sender’s number, such that if the SIM card is lost or the SIM card is duplicated, the attacker can use the victim’s account to perform transaction and also the USSD message that gets sent to the bank server is only encrypted between the mobile station and the base transceiver station. The message is in plaintext within the mobile operator’s network.
B. Pseudocode

The pseudocode for the algorithms are discussed in the following section,

a. Three party key agreement protocol using ECC
b. Signcryption scheme
c. RGB intensity based Randomized pixels with variable bits image Steganography

C. Three party key agreement protocol using ECC:

This protocol is used to obtain a session key which is used in signcryption scheme. It involves 2 phases, one is assignment phase and other is key exchange phase [5]. Assignment phase: In this phase, all the three parties C, B and T are agreed upon certain parameters, which are described in the following steps.

Step 1: Select prime finite F and check p>2^{192}
Step2: Choose the field element ‘a’ and ‘b’ which satisfies 4a^4 + 27b^2 ≠ 0 (mod p) and select base point G of order n
Step3: Publish (F_p, Elliptic curve over finite field F_p, Advanced Encryption Standard AES, Decryption function, AES Decryption function, Identity of Client and Bank server, Secret Octet string, One way hash function, Output of Key Derivation Function, Shared symmetric key, Tag for encrypted message, Logarithm of ‘a’, Concatenation algorithm).

Step4: Bank customer C and bank application server B register to trusted server T. Then, C, B and T generates private/public key pairs \(d_c, d_b, d_t\) respectively,
\(U_c = d_c \cdot G, U_b = d_b \cdot G, U_t = d_t \cdot G\)
where \(*\) represents point multiplication.
Step5: Select \(U_a\), a secure point known only to customer and bank server and it differs for each customer.

Key exchange phase: In this phase, here C and B, exchange some parameters through T, that parameter will be used for generating session key. This phase is further subdivided into three Subphases.

Subphases1 Client application(C)
Step1: Calculate \(K = d_c \cdot U_t = (K_{cx}, K_{cy})\)
Step2: Compute \(EC_c = \text{Encrypt}(ID_b)\)
Step 3: If Decryption Succeeds, then goto Step 4 Otherwise, T sends an authentication failure message to C
Step4: Compute \(K = d_b \cdot U_b = (K_{bx}, K_{by})\)
Step 5: Generate a random number \(r \in E(F_p)\)
Step 6: Compute \(EC_{tc} = \text{Encrypt}(r)\) and \(EC_{tb} = \text{Encrypt}(r, ID_c)\)
Step 7: T send \(EC_{tc}\) to C and \(EC_{tb}\) to B.

Subphases 2 Trusted server(T)
Step1: Receive \((ID_c, EC_c)\) from customer
Compute \(K = d_c \cdot EC_c = (K_{cx}, K_{cy})\)
Compute \(PT_c = \text{Decrypt}(EC_c)\)
Step 2: If Decryption Succeeds, then goto Step 3 Otherwise, T sends an authentication failure message to C
Step3: Compute \(K = d_b \cdot U_b = (K_{bx}, K_{by})\)
Generate an random number \(r \in E(F_p)\)
Step 4: Compute \(EC_{tc} = \text{Encrypt}(r)\) and \(EC_{tb} = \text{Encrypt}(r, ID_c)\)
Step 5: T send \(EC_{tc}\) to C and \(EC_{tb}\) to B.

Subphases 3 Session Key Generation
Step1: C receives \(EC_{tc}\) and B receives \(EC_{tb}\) from T
Step2: C compute \(K = d_c \cdot U_t = (K_{cx}, K_{cy})\) and B computes \(K = d_b \cdot U_t = (K_{bx}, K_{by})\)
Step3: Both C and B decrypts the received \(EC_{tc}\) and \(EC_{tb}\) using the keys \(K_{cx}\) and \(K_{bx}\) respectively.
Then C computes \(r = \text{Decrypt}(EC_{tc})\) and B computes \(r = \text{Decrypt}(EC_{tb})\)
Step4: Both C and B calculates \(SK = U_a \cdot r\) \(SK = (SK_x, SK_y)\)
where SK represents session key

D. Signcryption scheme

In the signcryption scheme it involves two phases[3], one is to generate the signed message and other is to verify signed message. The proposed scheme references the existing scheme[3] and the modified steps are highlighted bold in both the phases. Signcrypted text generation phase: In this phase, message is encrypted using (AES) encryption algorithm [3], it uses session shared symmetric key (EM) and for the encrypted message, sign is generated using private keys dc and db for C and B respectively.

Step1: Select random integer \(r_a \in E(F_p)\)
Step2: Compute \(R_c = r_a \cdot G\) where \(R_c = (R_{sx}, R_{sy})\)
Step3: Compute \(K_c = (r_a + d_c) \cdot U_b\)
\(R_{sx} = 2^{(pow/2)} + (R_{sx} \mod 2^{(pow/2)})\)
Step 4: Check $K_s = 0$, if $K_s = 0$ go to step 1 else
Compute $Z = H(K_{sx} \| ID_c \| SK_x \| K_{sy} \| ID_b)$
Step 5: Compute $KDF = H(Z \| \text{Counter})$ where
$KDF = EM || DM$
$CT = Encrypt(M)$
Step 7: Compute $x = H(CT \| R_{sx} \| ID_c \| DM \| R_{sy} \| ID_b)$ and
Compute digital signature $S = x * d_c - r_{sx} \pmod{n}$
Step 8: Send signcrypted text $(R_s, CT, S)$

Unsigncryption phase: In this phase, for the received signcrypted message, sign is verified using public keys $U_c$ and $U_b$ for C and B respectively. If sign verification succeeds, then encrypted message is decrypted using AES decryption algorithm. Otherwise message is discarded.

Step 1: Compute $K_c = d_b (R_s + * U_c) = (K_{sx}, K_{sy})$
Step 2: Compute $Z = H(K_{sx} \| ID_c \| SK_x \| K_{sy} \| ID_b)$
Step 3: Compute $KDF = H(Z \| \text{Counter})$
Step 4: Compute $x = H(CT \| R_{sx} \| ID_c \| DM \| R_{sy} \| ID_b)$
Step 5: If $S * G + R_s = x * U_c$ then accepts CT to decrypt, Otherwise Rejects CT
Step 6: Decrypt the encrypted message CT and plain text $M = Decrypt(C T)$

E. RGB Intensity based Randomized pixels with variable bits image steganography algorithm

This algorithm hides data inside an image based on Intensity of RGB channels and partition scheme [4]. To hide the data, random pixels are chosen based on $U_a$. For generating indicator sequence and for selecting partition scheme ‘r’ is used, which is sent by T.

Embedding phase:
Step 1: Generate an Indicator Sequence from ‘r’.
Step 2: Select random pixels based on ‘$U_a$’
Step 3: Generate partition scheme based on ‘r’ Choose number of data bits to store based on partition scheme and channel value
Step 4: Store the data bits
Step 5: Modify the other channel’s LSB, which is used while retrieving the data.

Extraction phase:
Step 1: Generate an Indicator Sequence from ‘r’
Step 2: Select random pixels based on ‘Ua’
Step 3: Determine the channel hiding data by checking LSB of two channels in a pixel other than an indicator
Step 4: Determine number of data bits hidden in the above determined channel based on partition scheme.
Step 5: Read the data bits

4. Results

The proposed system overcomes the security shortfalls in the existing system [1]. The proposed system provides security services such as availability, mutual authentication, non-repudiation, data integrity. The cost of each transaction in proposed system is less, when compared to GPRS banking. The minimum number of SMS required to perform a single transaction is two. The embedding capacity of the image (124 X 124) pixels is 30 times greater than the size of SMS packet. There are no major variations in the histogram of the image after data is embedded into it.

5. Conclusion & Future Work

In this paper, we have first presented the background of mobile banking system and follows design, implementation & results of a proposed system. The future works relates to the secure m-payments. Our proposed system acts as an initiative to MMS banking. In the near future, whole mobile banking system is transferred from partial MMS banking to complete MMS banking depending on mobile phones.

References