Elliptical Curve Cryptography in MobiCash

Akash Shetty, Yash Chapaneri, Tushar Ghalsasi, Shrikant Parte, Smita Patil

Atharva College of Engineering
University of Mumbai
Mumbai, india
akash.j.shetty91@gmail.com
yashtatsme@ymail.com
smitasukrut@gmail.com

***Abstract*—When we say Cryptography we usually refer to the study of techniques for ensuring the secrecy and authentication of the information we wish to share. Elliptic curve arithmetic can be used to develop a variety of Elliptic Curve Cryptographic (ECC) schemes including key exchange, encryption and digital signature. In this paper we have proposed a system for payment known as MobiCash that can be used in online cash transactions. We have shown how payment can be securely handled in MobiCash with the help of ECC. The principal attraction of Elliptic Curve Cryptography compared to other cryptography techniques is that it offers equal security for a smaller key-size, thereby reducing the processing overhead.**

***Index Terms*—Security; ECC; ECDSA; Hash algorithm; Authentication**

I. INTRODUCTION

Elliptical Curve Cryptography (ECC) is a public key cryptography. We have two pair of keys in Elliptical curve cryptography, public and private keys and a set of operations associated with the keys to do the cryptographic operations. Only the specific user knows about his private key where as the public key is distributed amongst all users taking part in this communication. At least some public key algorithm may have a requirement of a set of predefined constants to be known by all the devices which take part in this type of communication. An example of such constants is called ‘Domain parameters’. Unlike private key cryptography, public key does not have a requirement of any ‘shared secret key’ between the communicating parties, but it is much slower than the private key cryptography.

A newer approach to this is ECC, and it is considered as an extraordinary technique. It has a large exponential time system because of which it makes it difficult for people to break into the system. Security provided by a 1024 bit key of RSA is equivalent to 160-bit key of ECC which facilitates lower computer power. Hence for a given key size ECC offers considerably higher security. As a Consequence, for a faster implementation for a given level of security, the possibility of success increases with minimum size of key. So to run faster cryptographic operation we need compact keys. For ECC exponential operations, there are extremely efficient and compact hardware implementations that are available. offering potential reductions in implementation even beyond those due to the smaller key length alone. Not only ECC is emerged as an attractive public key crypto-system for mobile/wireless environments but also provides bandwidth savings. Miller and Koblitz proposed the use of elliptic curve in cryptography. Elliptic curve cryptography is not easy to break because it is difficult for attacker to understand it.

The domain parameters, the finite field representation, elliptic curve algorithms for field arithmetic as well as elliptic curve arithmetic makes the choice of elliptic curve deterministic. The security conditions under consideration determines the optimum selection of these parameters. In symmetric key cryptography single key uses for both encryption and decryption. In asymmetric key algorithm it however uses only for decryption of encrypted message. In asymmetric key cryptography, public key is used for message encryption and widely distributed for public. Elliptic curve cryptography is asymmetric key cryptography by nature.

II. ARCHITECTURE OF MOBICASH

A anonymous mobile payment system that supports principle security features, efficiency and fully anonymity for mobile users is called MobiCash system. When a customer is anonymous from bank and merchant we call this system as fully anonymous. e-cash system constructs this protocol there are some problems associated with the e-cash system that we try and eliminate with our protocol. The primary problem with the e-cash system being that the customer and merchant should hold accounts within the same bank which also supports e-cash payment system. The Other problem with the e-cash system is, if merchants’s bank is different with that of customer’s bank, the coins that are generated in merchant’s bank will not be accepted in customer’s bank and vice versa. So we say that an online mobile payment system providing a high level of privacy and anonymity to a mobile user is MobiCash. Preventing double spending and overspending of coins are the other additional features provided by MobiCash. In the following section, the overall architecture and approach of MobiCash is described. Six primary entities that are contained in MobiCash are:

1. The user (customer): He/she is the one who owns a mobile device. He/she, using MobiCash system purchases goods and services from the merchant.
2. The merchant (vendor): sells electronic goods or services to customer.
3. The digital wallet: A software containing software and code to support the interaction's with merchant and customer.
4. Trusted third party (TTP): A party that is certificate issuing authority for bank's signature and it is like clearing house for banks. Double spending and over spending is also checked by the TTP.
5. Customer's bank: Customer having account in a bank is the customer's bank. This signed coins are withdrawn from the customers account which is generated by the customer.
6. Merchant's bank: is the bank that merchant has an account in it and this bank deposits the received from the customer .

III. ARCHITECTURE

An electronic wallet containing digital coins mainly constructs this protocol. The wallet is used by the customer either when he wants to subscribes for the mobile service or when he wants to use it as a special subscription service with a monthly transaction fee. The wireless provider may install the electronic wallet on a mobile device.

Before any development in transaction, A secure channel is established by the TTP and the bank. This allows the TTP to generate digital certificates for the banks, in order to verify the bank signatures.

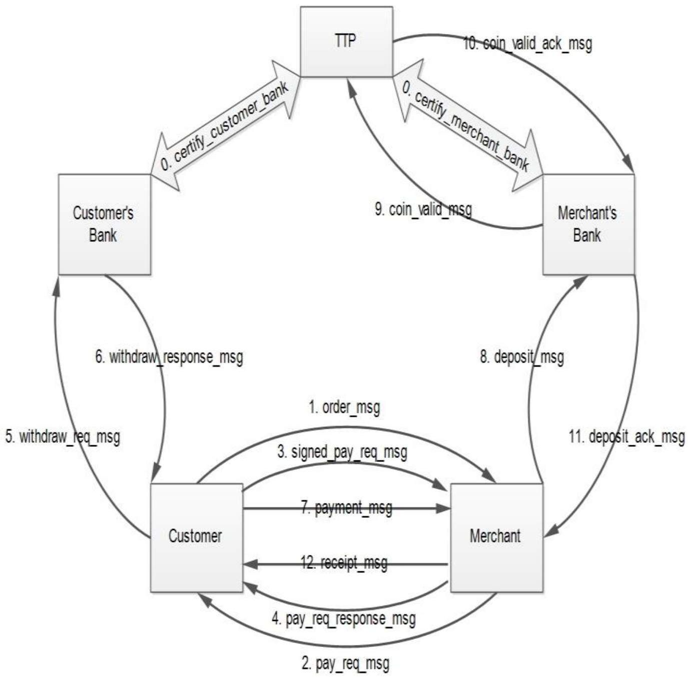

Figure 1: MobiCash Architecture

1. Customer goes to the merchant's website and order's the product he wants.
2. A 'pay_req' message is generated by Merchant and then it is sent to the customer's electronic wallet. This message contains The details about the amount, the currency to be used, order, and the current time comprises of the message that is sent.
3. The customer signs and generates 'signed_pay_req' message Upon receiving 'pay_req' message and sends it to merchant.

Customer →Merchant: signed_ Pay_req _msg
Signed_Pay_req_msg={purchase_information |$Sig_{customer}$}$p_{kmerchant}$ purchase_info={Total_amount|currency|product_description|timestamp} $Sig_{customer}$= {H(Total_amount|currency|product_description|timestamp)} $S_{kcustomer}$

Figure 2: pay_req message.

4. The merchant generates 'pay_req_response' message, on receiving 'signed_pay_req' message, he/she and then sends it to customer.

Merchant → Customer: Pay_req_response_msg
Pay_req_response_msg= {purchase_info|$pk_{merchant\_bank}$|merchant_bank_ID| merchant_account_ID | $Sig_{merchant}$ }$P_{kcustomer}$ Purchase_info={total_amount|currency|product_description|timestamp} $Sig_{merchant}$={H (purchase_info|$pk_{merchant\_bank}$|merchant_bank_ID|merchant_account_ID)} $S_{kmerchant}$

Figure 3: pay_req_response.

5. The customer with the help of his electronic wallet installed on his mobile device generates digital coins, on receiving the 'pay_req_response' message. Then he blinds coins via With the help of blind signature algorithm, he then blinds the coins and sends customer's bank, withdraw_req'.

Customer→Customer,s Bank :Withdraw_req_msg
Withdraw_req_msg ={blinded_coins |Total_amount |currency|$Sig_{customer}$ }$P_{k\_customer's\ bank}$ $Sig_{customer}$ =H (blinded_coins | Total_amount | currency)$S_{k\_customer}$

Figure 4: withdraw_msg.

6. Customer's account is debited by the customer's bank on receiving these coins, and the coins are blindly signed. After that it sends signed blinded coins to the customer alongwith a 'withdraw_response' message.

Customer's Bank→ Customer: Withdraw_response_msg
withdraw_response_msg={signed_blinded_coins|$Cert_{customer's bank}$|$Sig_{customer's bank}$}$P_{kcustomer}$
$Sig_{customer's\ bank}$ = {H(blinded_coins | $Cert_{customer's\ bank}$)}$S_{kcustomer's\ bank}$

Figure 5: withdraw_response_msg.

7. The customer starts unbinding the signed coins he received, and sends the merchant a 'payment' message.

Customer → merchant: Payment_msg
Payment_msg={payment_info|{signed_coins|$Cert_{customer'sbank}$|H (payment_info)}$P_{kmerchant's\ bank}$ |$Sig_{customer}$ }$P_{kmerchant}$
Payment_info= {purchase_info| merchant_bank_ID| merchant_account_ID | H (payer_code)}
Purchase_info= {Total_ amount| currency| product_ description |timestamp }
$Sig_{customer}$={H (payment_info|{signed_coins|$Cert_{customer'sbank}$| H (payment_info)}$P_{kmerchant's\ bank}$)}$S_{kcustomer}$

Figure 6: payment_msg.

8. Merchant signs the coins and, then forwards it to the merchant's bank as a part of a deposit request, after he receives 'payment' message .

Merchant → merchant's bank: Deposit_msg
Deposit_msg = {{payment_msg} |$Sig_{merchant}$ } $P_{kmerchant's\ bank}$
$Sig_{merchant}$ = {H (payment_msg)}$S_{kmerchant}$

Figure 7: deposit_msg.

9. The merchant's bank generates 'Coin_validation', verifies merchant's signature, message and then sends it to the TTP where it undergoes an online verification and double spending (over spending) prevention, After the merchant's bank receives 'deposit' message.

Merchant's bank→ TTP: Coin_validation_msg
Coin_validation_msg = {signed_coins |$Sig_{merchant's\ bank}$}P_{kTTP}
&
$Sig_{merchant's\ bank}$ = {H (signed_coins)}$S_{kmerchant's\ bank}$

Figure 8: coin_validation_msg.

10. The following operations are done by TTP:
    a. Signature verification of customer's bank on coins for validation.
    b. Expiry date of coins is checked.

TTP : check coin validity
1) check the $sig_{customer's\ bank}$ by using $Cert_{customer's\ bank}$
2)check the Exp_Time
If step 1 and 2 were true then go to step 3

Figure 9: coin_validity.

    c. Database is searched, this ensures that a coin is not spent twice or more, the transaction will be aborted if the coins are found in the database else if coins are not found they are valid and are recorded in the database if coins do not exist in the database.

TTP : double spending prevention
3)searches coin in its data base
If the coin exist in database then
result="False"
Else
result="True"

Figure 10: double spending.

    d. The merchant's bank receives a 'Coin validation acknowledge' that is generated and signed by the TTP.

11. Upon receiving the' Coin validation acknowledge' message, if the result was "true", the merchant's bank credits merchant's account. In addition an indication of success is returned to the merchant.

TTP→ merchant's bank: Coin_validation_ack_msg
Coin_validation_ack_msg= {result|Sig_{TTP} }$P_{kmerchant's\ bank}$
Sig_{TTP} = {H(result)}S_{kTTP}

Figure 11: Coin_validation_message.

12. The customer then receives the purchased items and its receipt that is apparently sent by the merchant to finish of the process.

Merchant's bank→merchant: Deposit_ack_msg
Deposit_ack_msg= {result|Total_amount|$Sig_{merchnat's\ bank}$}$P_{k\ merchant}$
$Sig_{merchnat's\ bank}$ = {H(result|Total_ amount)}$S_{kmerchant's\ bank}$

Figure 12: deposit_ack_msg.

IV. IMPLEMENTATION

Koblitz and Miller in 1985, independently proposed Elliptic curve cryptosystem. Bandwidth saving, computational power small key size and integrated circuit space are limited, such as smart cards, and wireless devices [3,7 and 8] are some of the features of this system. With the help of ECC we have implemented the proposed system in this section.. As an example we mention the Elliptic Curve ELGamal Cryptosystem (ECEC) and Elliptic Curve Digital Signature Algorithm (ECDSA).

A. ECDSA Key Generation

The following procedure shows how the users should generate the public and private keys [3,7 and 8]:

1. Let us select an elliptic curve 'ec' over a finite field named as F(p). let us now assume m is a

large prime no, then the no of points that lie on 'ec' should be divisible by m.

2. Now let us Choose a point P=(a,b)∈ F(p) of order m.

3. Now let us Choose randomly an integer i∈[1, n-1].

4. Compute Q=iP.

5. The public keys thus for users are (Q,m,P,Ec), and the private key is i..

B. ECDSA Signature Generation

In order to sign a message, let us take an example *'Pay_req_response_msg'*, merchant with an EC domain parameters that of I and the key pair (i,Q) will take the following steps for ECDSA signature generation(see figure 3).

1. Let us select a random integer say k, 1≤ k ≤ n - 1.
2. Compute the range of kP = (a, b).
3. Compute the following steps and attain values:
   • r ≡ a (mod m*)*. If *r = 0*, then go to the initial step.
   • k−1 (mod m); and h = *SHA-1(*m2|m4*)* of m3 and convert this bit string to an integer *e*.
4. Compute s ≡k−1(e + inr) (mod m). If *s = 0*, then go to the initial step (*dm* is merchant's private key). Merchant's signature for the message m3 is (r, s).

C. Elliptic curve Encryption by the ElGamal algorithm

The following procedure shows how merchant should encrypt his message m3 by customer's public key *(Qc)* and generate his cipher text C1[3,7 and 8]:

1. Consider m3 as a point.
2. Select a positive integer number j.
3. Compute a3=jP and b3=m3+jQc and then obtain C1=(x3,y3*)* .

In this step merchant sends ciphertext C1 to customer.

D. Elliptic curve Decryption by the ElGamal

The customer on receiving cipher text C1, decrypts it by using its private key (i) as follows[3,7 and 8]

M3=b3-ica3

E. ECDSA Signature Verificaion

To verify Merchant's signature (r, s) on M3, the customer must obtain an authentic copy of merchant's domain parameters D and associated public key Q . Verify that r and s integers over [1 , n-1].

Compute the message digest *h = SHA-1*(m2|m4) of the message m3 and convert this bit string to an integer *ec.*

Compute the following steps [3,7 and 8]:

- *W≡S^-1(mod m.)*
- ui≡ecw(mod m) and uj≡ rw (mod m).
- X=(a,b)= uiP+ujQm.
- If *X=infinity* then return ("Invalid signature");
- Represent x∈*F* as an integer x'∈ Z.
- v≡a(modm*)*.
- If v=r then return ("valid signature") else return ("invalid signature).

V. RESULTS

A. The payment

A customer generates a secret 'Payement_code' when he purchases any item. After this the system generates H(payement_code) which is the hash of the payment code. The customer in order to prove the merchant's bank about his payment , can also include this payment information when the merchant deposits the coins that are generated and thus confirm the payment. After the merchant deposits the payment that contains H(payement_code) the bank will records this and when the customer reveals a payement_code that he earlier created and recorded, the bank will be assured that the customer is willing to and has made the payment.

B. Double spending

In this protocol we have proposed a system wherin TTP that has a repository which records the valid coins in it. So as a result, when merchant's bank receives the coins that are generated, it forwards them to TTP. And thus As stated in step 8 of the architectural protocol, TTP does operations to validate and prevent double spending.

C. Anonymity

The central objective of the paper is that the customer should remain anonymous and in the system the bank is unable to see the serial number of coins, since the serial number of coins is blinded and sent to the customer's bank, and thus the customer remains anonymous.

VI. CONCLUSION

We have seen that in the present age Mobile and electronic commerce has been the hot topic of application and research, and the E-cash system is the key technology and backbone of electronic commerce. With the medium of this paper, we have introduced the MobiCash system where in we have shown that payment in M-commerce can indeed have digital cash and work swiftly. Also the central objective of the paper Mobicash has designed to work online for mobile devices to support security and fully anonymity for mobile users is which is extremely necessary when dealing with money transactions. We have tried to eliminate some problems of e-cash system that were earlier prevalent in the system that Both customer and merchant must have accounts at the same e-cash bank and also that Coins obtained from one bank will not be accepted by another; with the help of TTP which eliminates both these issues.

REFERENCES

[1] D.Chaum ,"Blind Signature for Untraceable Payments" In Advances in Cryptology-CRYPTO' 82, pp. 199-203,1983.
[2] W.Chung ,"Mobile Commerce Security and payment methods", Auburn University,Usa, IDEA GROUP Publishing,2005.
[3] M.Y.Rhee,"Internet Security Cryptographic Principles, Algorithms' and Protocols" press by John Wiley, 2003.

[4] H.Marko and H.Kostantin and T.Elena ," Utilizing national public key infrastruature in mobile payment systems",press in Elsevier, 28 march,2007.
[5] A.Nash , W.Duane and C.Joseph,"PKI Implementing and Managing Esecurity" RSA press,2001.
[6] A.Ranjit , M.Ravi and V.V.Kummar, "MobiCoin: Digital cash for MCommerce", springer-verlag ,pp. 441-451, 2004.
[7] N.Sklaovs and X.Zhang, " Wireless Security and Cryptography" ,CRC press, 2007.
[8] T.S.Denis and S.Johnson," Cryptography for Developers", Syngress press,2007.
[9] A.Silberschatz," Database System Concepts (fourth edition)"MCGraw-Hill company, 2001..