ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0055 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

The Generalised Secured Mobile Payment System Based on ECIES and ECDSA

Ehab M. Alkhateeb
Faculty of Science and Information Technology: Al Zaytoonah University of Jordan
Amman, Jordan
ehabalkh@gmail.com

Mohammad A. Alia
Faculty of Science and Information Technology: Al Zaytoonah University of Jordan
Amman, Jordan
dr.m.alia@zuj.edu.jo

Adnan A. Hnaif
Faculty of Science and Information Technology: Al Zaytoonah University of Jordan
Amman, Jordan
dr.adnan_hnaif@zuj.edu.jo

***Abstract*— Mobile payment system is defined as an electronic payment method, also it is defined as mobile money transfer and mobile wallet. Since mobile payment has been generated to be an attractive alternative for the traditional payments systems such as credit cards. In this paper Elliptic curve cryptography is used to mobile payment system. Meanwhile, the proposed mobile payment system includes three main processes: Authentication process, Member recognition process, and Payment process. Moreover, Elliptic Curve Integrated Encryption Scheme ECIES and Elliptic Curve Digital Signature Algorithm ECDSA cryptographic protocols have been applied to enhance the security of the proposed mobile payment system. However, the proposed system is secure, easy and straightforward payment process. As well as, USSD technology is used in this system for PIN authentication process with high security performance.**

***Keywords*— ECC, ECIES, ECDSA, Mobile Payment System, and Cryptography.**

I. Introduction

Mobile payment is defined as payment for products or services between two parties for which a mobile device, such as a mobile phone, plays a key role in the realization of the payment [1]. Nowadays mobile phones are spreading widely through social communities, and becoming a replacement for laptops and desktop PCs. The user demands for convenient and intelligent ways in which to make payments for goods and services using a mobile phone is creating exciting opportunities for those organizations that are part of the mobile payment ecosystem [2]. However, Mobile payments facing critical security issues with the rise of identity fraud, and illegal access to confidential data, such as credit card details. Furthermore, the cryptanalysis and attacking, protocols speed, and performance evaluation are the core elements in building a secure mobile payment system. Therefore, this paper focuses its attention on these concerns by presenting a mobile payment which is based on public key cryptography. The assessment of security for the proposed mobile system is based on the strength of proposed cryptographic algorithm, the selected key size, the performance and the speed of the proposed system. This approach is rather a replacement or a merge for classical way to pay using credit card and the current mobile payment methods.

II. Related works

A. Cryptography

Cryptography is a cornerstone of the modern electronic security technologies used today to protect valuable information resources on intranets, extranets, and the Internet. Cryptography is the science of providing security for information [3]. Cryptography have many algorithms that can be categorized into two main types based on the nature of key, namely secret and public keys. The secret key or non-public key cryptosystem only need one key(secret key) to encrypt and decrypt the data between the sender and the recipient, while public-key cryptosystems comes in more difficult approach, it consists of two keys, the public key which is used to encrypt the data and private key for decryption . Public key based on key exchange protocol rises above the difficulties faces by the secret key cryptosystem. This is because key management is much easier with the help of a key exchange protocol such as Diffe-Hellman [4].

B. RSA (Rivest, Shamir, Adelman) protocol

The RSA protocol [5] is one of most widely used public key cryptography algorithms. The algorithm was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm relays on large integers and prime testing, its mathematical basis is the Euler theorem, the security of RSA depends on the difficulty of factoring larger integer [6]. However, RSA have many disadvantages, and with time passing its being replaced with more efficient algorithms, the following are some disadvantages of RSA cryptosystem [6]:

- Fake public key.
- Complexity of key creation.
- Security need to be proofed.
- Slow of the speed.

C. RSA Digital Signature (RSA DS)

In the RSA algorithm for encryption and decryption process uses public key to encrypt and private key to decrypt as mentioned previously. The RSA Digital Signature (RSA DS) uses the private key to generate a signature and the public key is used to verify that signature [6].

D. ECC Cryptography

Elliptic Curve Cryptography (ECC) nowadays having a lot of attention, due to its small key size for encryption, decryption and digital signature, beside entered a wide use in 2004 and 2005. ECC was discovered by Nael Koblitz [7], and Victor S.Miller [8]. The ECC schemes are public-key mechanisms that provide the same functionality as RSA schemes. However, their security is based on the hardness of a different problem, namely the Elliptic Curve Discrete Logarithmic Problem (ECDLP) [9]. The adoption for ECC makes it a competitive to RSA, since it can reach the same security level with smaller key size, smaller key size means less computation time and high performance speed. ECC can be used in variuous areas, as encryption algorithm like ECIES which is a replaceble for RSA cryptosystem, or key exchange protocol such as ECDH, or as a digital signature such as ECDSA which recently being used intensively through the internet to provide integrity and non repudation for messages. Elliptic curve protocols depend on ECDLP, which it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible.

E. ECIES cryptosystem

One of the most efficient encryption and decryption based on elliptic curve is ECIES. ECIES is a public-key cryptosystem [10]. ECIES proposed by Abdalla, Bellare, and Rogway in [11] and [12]. As its name properly indicates, ECIES is an integrated encryption scheme which uses the following functions [10]:

- Key Agreement (KA): Function used for the generation of a shared secret by two parties.
- Key Derivation Function (KDF): Mechanism that produces a set of keys from keying material and some optional parameters.
- Encryption (ENC): Symmetric encryption algorithm.
- Message Authentication Code (MAC): Data used in order to authenticate messages.
- Hash (HASH): Digest function, used within the KDF and the MAC functions

To apply encryption and decryption using ECIES between Alice (sender) and Bob (receiver) must do the following [3]:

a. The previous Cryptographic functions.

b. Elliptic Curve domain parameters (p,a,b,G,n,h) for a curve over prime field or ($m,f(x),a,b,G,n,h$) for a curve over binary field.

c. Keys generation:
   - Bob's public key : V (Bob generates it as follows: $V = v.G$, where v is the private key he chooses at random: v € [1, n-1]
   - Alice’s public key: U (Alice generates it as follows:
     U=$u.G$, where u is the secret key she chooses at randon: u € [1, n-1]
   - Optional shared information (parameters): $S1$ and $S2$.

d. Encryption: To encrypt a message m, Alice does the following:
   1. Derives a shared secret: $S = Px$, where P = (Px, Py) = $u.V$ (and P $\neq$ 0)
   2. Uses KDF to derive a symmetric encryption and a MAC keys: $K_{ENC} \| K_{MAC}$ = KDF($S \| S1$);
   3. Encrypts the message: c = ENC (K_{ENC}; m);
   4. Computes the tag of encrypted message and $S2$: d = MAC(K_{MAC}; $c \| S2$);
   5. Outputs $U \| c \| d$.

e. Decryption: to decrypt the ciphertext, Bob does the following:
   1. Cryptogram $U \| c \| d$.
   2. Derives the shared secret: S = Px , where P = (Px, Py) = v.U (it is the same as the one Alice derived because $P = v.U = U.v.G = u.V.G = u.V$, or outputs failed if P = 0;
   3. Derives keys the same way as Alice did: $K_{ENC} \|$ K_{MAC} = KDF(S||S1);
   4. Uses MAC to check the tag and outputs failed if d $\neq$ MAC(K_{MAC}; c||S2);
   5. Uses symmetric encryption scheme to decrypt the message m = ENC $^{-1}$(K_{ENC}; c)

Figure 1 shows ECIES encryption and decryption process between Alice, and Bob.

H. End to end encryption (E2EE)

The E2EE is a concept used to secure data on flight from one device to another, in a pure networking it means to secure data between two endpoints unlike the client-server architecture. Figure 2 shows one of the variants E2EE namely POS to Acquirer Encryption (P2AE) where data are being encrypted before it being sent to the bank acquirer [14].

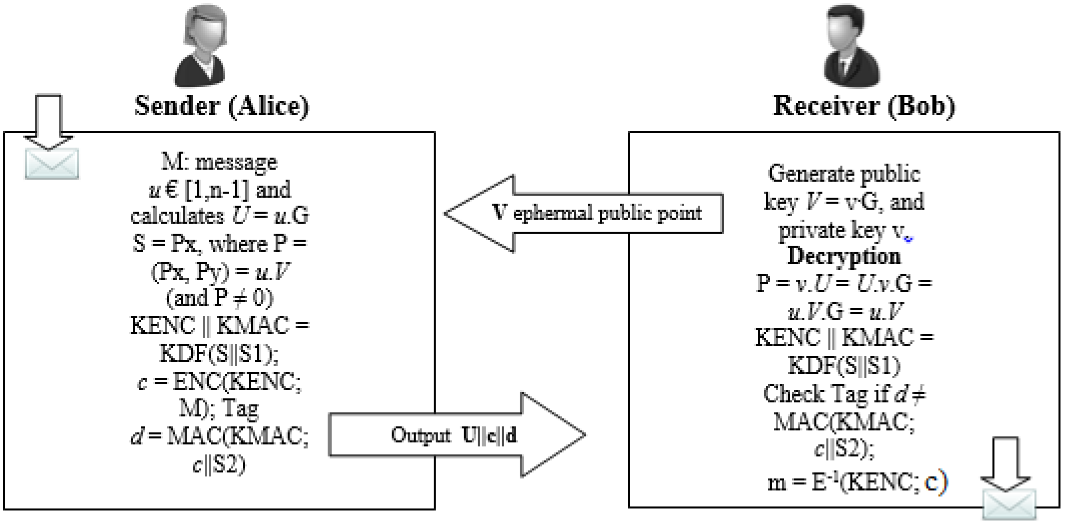


Figure 1 ECIES encryption and decryption process

F. Elliptic Curve Digital Signature Algorithm (ECDSA)

The ECDSA is a variant of the Digital Signature Algorithm (DSA) that operates on elliptic curve groups, for sending a signed message from A to B, both have to agree up on Elliptic Curve domain parameters. The domain parameters are defined in section Elliptic Curve Domain parameters. Sender 'A' have a key pair consisting of a private key *dA* (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key *QA* = *dA* * *G* (*G* is the generator point, an elliptic curve domain parameter) [3].

G. Identity-Based Cryptography (IBC)

The identity based concept depends on user's identifier information, such as phone number, e-mail, IP address etc., which replaces the digital certificates and to use public key for encryption or signature verification. Thus, this reduces the cost dramatically for establishing and managing the public key authentication framework known as Public Key Infrastructure (PKI) [13]. However, IBC facing major challenges as the following:

- Key ESCROW problem.
- Authentication is not sufficient.
- No support for certificates.

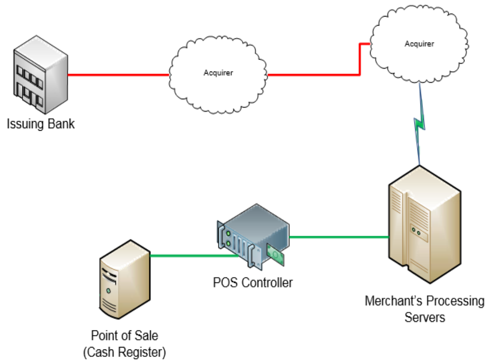


Figure 2 P2AE model [14].

I. USSD technology

The USSD technology is commonly used in banking, mobile polling, security systems and education, USSD is standard GSM technology supported by all GSM handsets. It is session-based and supports longer message content. Secure and cost-effective, sessions can be initiated by both end-users and enterprises [15]. USSD messages are more secured than SMS, data are protected when transmitted with default GSM security, and can be categorized into two types:

1. Push messages: Message received to the user that acquires a response.
2. Pull messages: Message initiated by the user by dialing then to acquire response.

USSD can be used for security improvements for banks or any other service that requires authentication, such as PIN entrance or one time password for a transaction authentication [16].

III. MOBILE PAYMENT SYSTEMS

There have been a number of deployments of mobile payments worldwide across the spectrum of proximity and remote payment. The following are some of these deployments:

A. A secure mobile payment service (SEMOPS)

Among the popular mobile payments systems, SEMOPS is a mobile payment solution that is capable of supporting micro, mini payments, and a universal solution, being able to function in any channel, including mobile, Internet and Point of Sale (POS), SEMOPS depends on securing transaction data by using RSA cryptosystem [17]. However, SEMOPS uses data center to control the flow of data, which requires many cooperate processes with different mobile operators and banks to accept the transaction. Thus, making it a much higher cost.

B. Google Wallet

Google Wallet is a popular mobile payment system developed by Google that allows users to store different payment cards in their mobile phones, Google Wallet uses NFC-technology to initiate payments on any pay-pass terminal at checkout [18]. NFC-technology is provided by verifone, and also provide E2EE protecting customer confidential data on transmittion.

C. Heartland mobile payment

Heartland apprach the same as in google wallet, both requires an NFC, and a wallet at POS, also the payment process steps almost the same. However, Heartland depend on Voltage security which provide an identity-base encryption, which also adopted by others mobile payment systems for E2EE to protect cardholder and sensitive authentication data throughout the payment acquiring (e.g. bank) network [19].

D. Paybox

Paybox a mobile payment system founded in 1999, Paybox spread in many Europe countries. Paybox Germany bank-centric mobile payment system connects with different banks to authorize mobile payments through customer bank accounts [20]. Paybox McDonalds in France have the same approach as in Google wallet, which depend on VeriFone, as part of its security is VeriShield, VeriShield provides robust security by protecting mobile payment transaction details on POS at rest or transmit through E2EE between merchant and payment processor or bank which uses RSA PKI relies on digital certificates as well as public and private cryptographic keys to secure information exchange, the unencrypted data will never exposed to thieves [21].

E. ECC for securing payment system

This mobile payment system is based on ECC cryptography, the Diffie-Hellmann key exchange protocol DHKEP is implemented by sending the public key of each party over insecure channel, also digital signature is used for authentication. The Cryptographic application used covers one of the mobile payment security requirements framework elements, namely: the application service provider which guarantees a secure end-to-end Communication and non-repudiation [22]. However, Elliptic Curve Diffie Hellman ECDH protocol lacks authentication. Thus, digital signature algorithm is used to overcome this issue.

F. Mobile payment method based on RSA

This system is based on RSA cryptosystem, also adopt SMS technology for PIN entrance [23]. However, RSA as mentioned previsoulsy facing many problems. Furethermore, SMS technology also facing problems related to security such as Identity fraud and man in the middle attacks.

G. Challenges facing current mobile payments

Mobile payments systems facing many challenges as mentioned previously, the main challenges can be classified as the following:

- Security :
  a. Cryptographic services lacks efficiency, and authentications needed.
  b. Vulnerability of many attacks, such as identity fraud or man in the middle attacks.
- Low supported phones: Low support of applied technologies such as NFC.
- Extra hardware or software for the customer.

IV. THE PROPOSED SYSTEM

The proposed secured mobile payment system (SMPS) mainly divided into three process, these are:

1. Authentication Process.
2. Client Recognition Process.
3. Payment Process.

The following are stakeholders of the proposed system:

- Customer.

- Merchant : act as mediator between the customer and the bank
- The Bank: Central point in the mobile payment, acting on behalf of payment processor, and manages the payment operation.
- Mobile Operator: act as service provider.

A. Authentication (Getting Service) Prcoess

The first process in the proposed system is the authentication process, where the bank, mobile operator takes the main role in this process, before the client can be able to use this service, the client must first register. As illustrated in Figure 3 step 1 the client meet with the bank employee and request the service, the bank employee checks if the client did provide the mobile phone number that he or she will use at the time of payment, if the number was not provided through their account, the bank employee will ask for it. As shown in Figure 3 step 2 the bank should acknowledge the mobile operator to authorize this service through sending the customer information. The mobile operator will respond with a notification message to the client of a successful registration after approving this service as show in Figure 3 step 3. The bank will complete the registration by giving the client the PIN as shown in Figure 3 step 4. Following these four steps the bank will generate a computed ECIES public-key *Bpu1*, and private-key *Bpv1*, also a computed ECDSA public-key *Bpu2*, and private-key *Bpr2* (refer to Figure 3 step 5) and then pass the two public keys *Bpk1*, and *Bpk2* to the market server (refer to Figure 3 step 6). Following that the market will generate a computed ECIES public-key *Mpu1*, and private-key *Mpr1*, also a computed ECDSA public-key *Mpu2*, and private-*Mpr2* (refer to Figure 3 step 7) and then pass the two public keys *Mpu1*, *Mpu2* to the bank (refer to Figure 3 step 8). At this stage the customer can shop in the market and pay through the mobile payment service.

B. Client recognition process

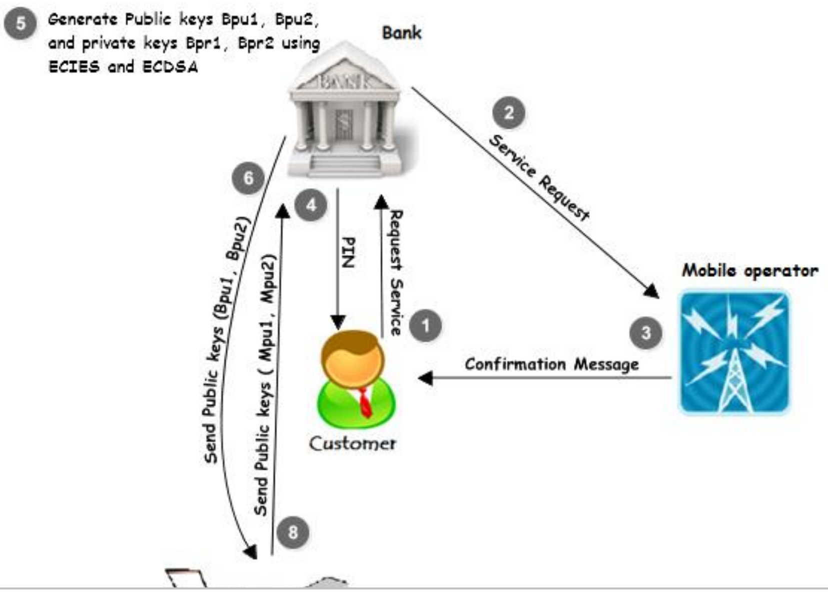
When the client enters the market for shopping (figure 4 step 1), the market server should test the customer either registered to the service or not (figure 4 step 2 and 3). If so a notification message will be sent to acknowledge that he or she can pay through the mobile payment service (figure 4 step 4). Otherwise, the customer will be ignored (figure 4 step 5).

C. Payment process

The final process in the proposed system is the payment process. The payment process divided into two phases, these are the payment phase and the payment confirmation phase.

- Payment phase:

The payment process starts when the client would like to buy


5
Generate Public keys Bpu1, Bpu2, and private keys Bpr1, Bpr2 using ECIES and ECDSA
Bank
BANK
2
Service Request
Mobile operator
6
4
PIN
Request Service
1
3
Send Public keys (Bpu1, Bpu2)
Send Public keys (Mpu1, Mpu2)
Confirmation Message
Customer
8

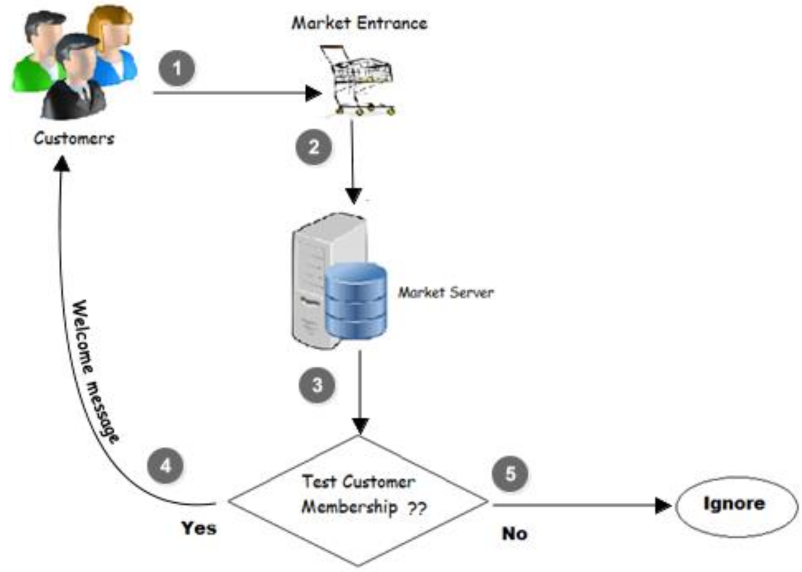


Figure 4 Client recognition process.

via mobile phone. First, the client have the choice to pay through mobile phone, Credit Card, or in cash as shown in figure 5 step1. At this stage if the client chooses the mobile payment, the client must pass the mobile phone number to the merchant to start processing the payment (figure 5). After that the following steps will be implemented securely, the customer membership will be checked through market server (Figure 5 step 2). If the client was a member, then the following steps will take place. The market server should contact the bank to provide the client information and the market information, the client information includes the amount to pay, and mobile phone number, while the market information includes the market ID and password, meanwhile the bank should make sure that the amount is available as shown in figure 5 step3.

Therefore, the ECIES, and ECDSA algorithms are implemented to encrypt and authenticate the customer information alongside the market information, this can be done using the bank public keys (*Bpu1*) to ciphertext (encrypted text) and using the merchant private key (*Mpr2*) to generate a signature. After that all these information will be sent to the bank. When the bank receives the ciphertext (encrypted text) two cases will be implemented. First, the ECIES is implemented to decrypt the data using bank private key (*Bpr1*), secondly the ECDSA is implemented to verify the signature using merchant public key (*Mpu2*) as shown in figure 5 step 4. After a successful implementation for the previous two cases the bank tests for valid amount and also checks for valid merchant ID and password as illustrated in figure 5 step 5, then upon successful checking process the bank should send the result to the mobile operator. After that, if the amount is valid as shown in figure 5 step 6 the mobile operator sends a USSD message to the client for payment confirmation, the client should respond to the message with the PIN as shown in figure 5 step 7 and 8. In figure 5 step 9 shows the deduction of the amount from the mobile operator to the bank.

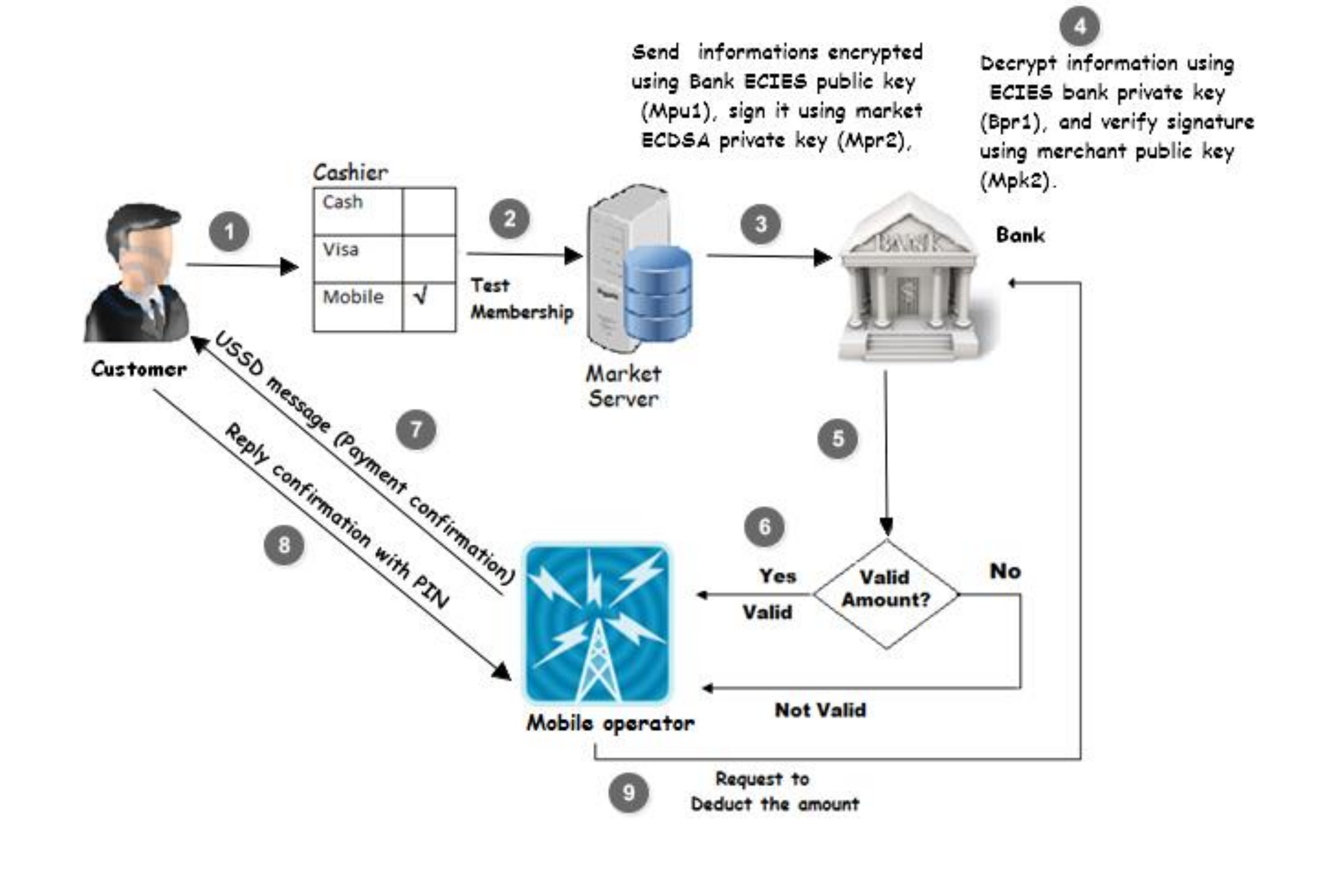

Send informations encrypted using Bank ECIES public key (Mpu1), sign it using market ECDSA private key (Mpr2),
4
Decrypt information using ECIES bank private key (Bpr1), and verify signature using merchant public key (Mpk2).
Cashier
Cash
Visa
Mobile √
1
2
Test Membership
3
Bank
Customer
Market Server
USSD message (Payment confirmation)
7
Reply confirmation with PIN
8
5
6
Yes
Valid
Valid Amount?
No
Not Valid
Mobile operator
9
Request to Deduct the amount

Figure 5 Payment phase.

- Payment Confirmation phase:

Following the payment phase it comes the payment confirmation phase in the payment process as shown in figure 6. This phase start through sending a report confirming that the payment process is executed successfully from the bank to the merchant server (figure 6 step1). This report is encrypted by implementing the ECIES algorithm using the merchant server public-key (*Mpu1*), and signed using the bank private-key (*Bpr2*).

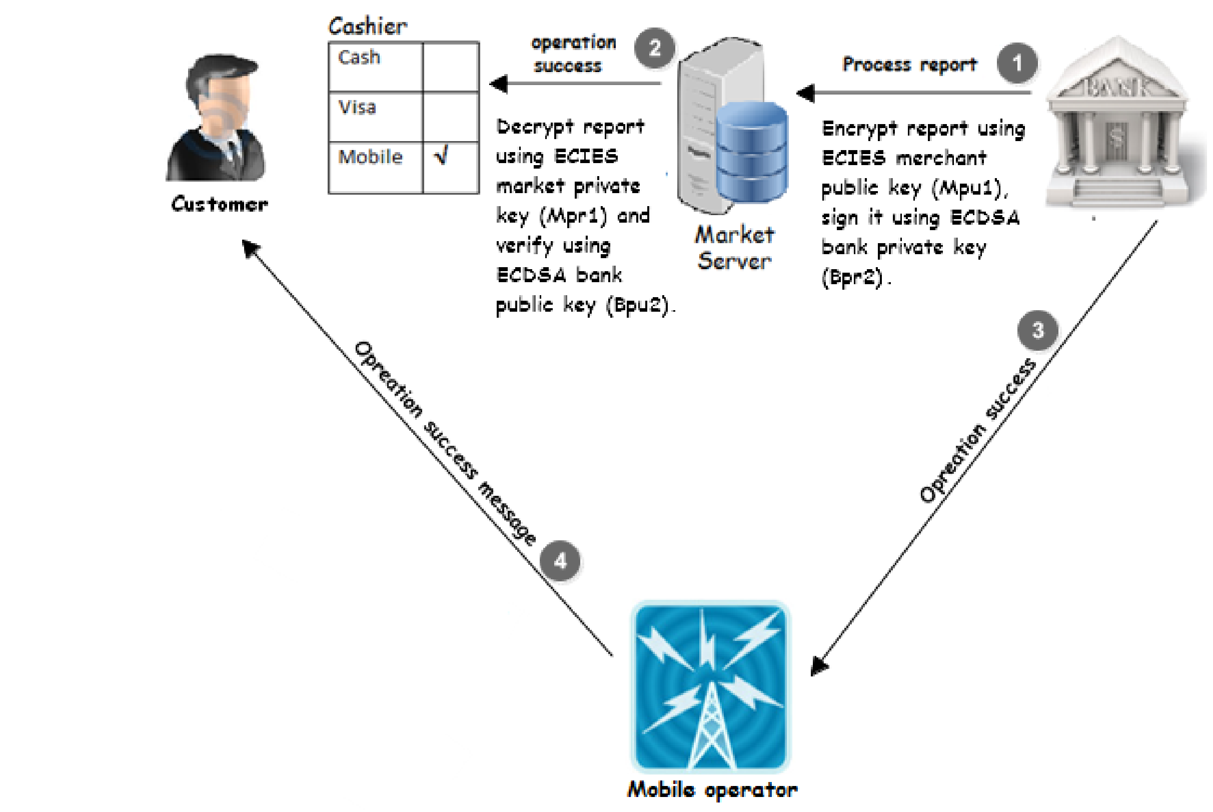

Cashier
Cash
Visa
Mobile
√
Customer
operation success
2
Decrypt report using ECIES market private key (Mpr1) and verify using ECDSA bank public key (Bpu2).
Market Server
Process report
1
Encrypt report using ECIES merchant public key (Mpu1), sign it using ECDSA bank private key (Bpr2).
BANK
3
Opreation success
Opreation success message
4
Mobile operator

Figure 6 Payment confirmation phase

D. The advantages of the proposed system :

The following are main advantages of the proposed system:

- Secured under theft, the PIN is secured using GSM channel through USSD push message. USSD does not store messages in the phone like SMS which rises the risk of fraud. Furthermore, it’s secured from Man in the Middle attacks.
- Provide high security level with high performance using ECIES and ECDSA applied on end-to-end encryption between merchant and the bank.

The market server receives the encrypted report and decrypt it by applying the ECIES algorithm using the market private-key (*Mpr1*) and verify signature using the bank public-key (*Bpu2*) as shown in figure 6 step 2. At this stage, the bank sends an operation success notification to the mobile operator (figure 6 step 3). Lastly, the mobile operator sends an operation success message to the client (figure 6 step 4).

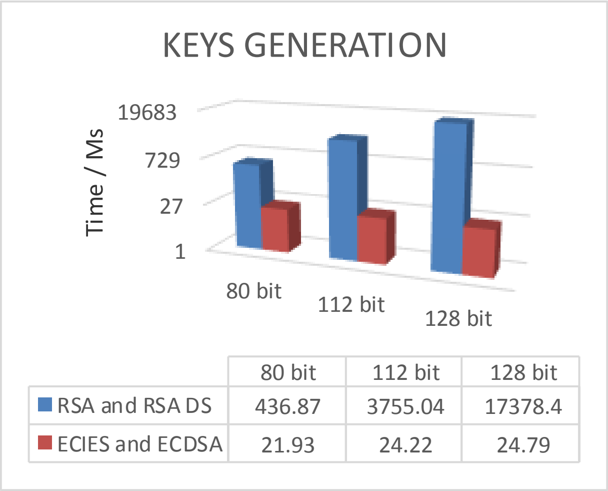
- No extra hardware or software needed for the customer in order to use the service.
- Wide range support of mobile phones, classical and smart ones.
- Easy and straightforward payment experience for customers.
- A replacement for the traditional credit card payments.

TABLE I. MBILE PAYMENT CHARACTERISTICS

Payment systems	**Characteristics**		
	Cryptographic algorithms	***Possibility of Identity Theft***	***Extra Hardware requirements***
Google Wallet	RSA and RSA DS	High	Yes
SEMOPS	RSA and RSA DS	High	Yes
Heartland	Boneh Franklin (IBC)	High	Yes
paybox	RSA	High	Yes
ECC for securing mobile payment system	DHKEP	-	-
Payment method based on RSA	RSA encryption	High	No
Proposed system (SMPS)	ECIES and ECDSA	Secured	No

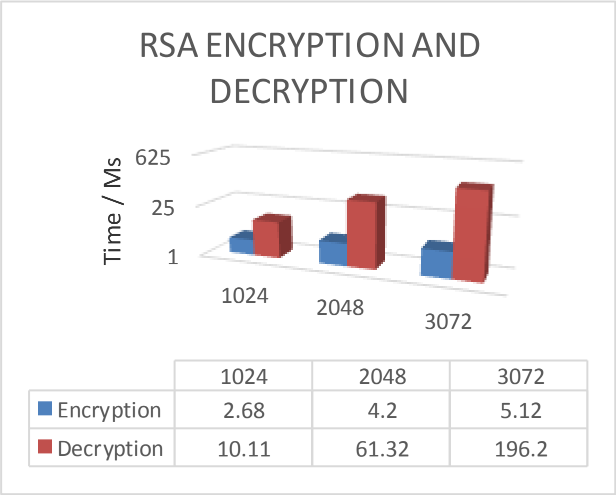
V. EXPERMINTAL RESULTS

In this section we discuss our experimental results of SMPS by applying RSA, ECIES, and ECDSA cryptosystems. The code implemented in windows 7 environment with Intel core due processor, using flexiprovider, bouncycastle APIs and NetBeans IDE. Flexiprovider is a powerful toolkit for the Java Cryptography Architecture (JCA/JCE) [24]. The implementation includes keys generation process and encryption, decryption processes and sign, verify processes. Figure 7 shows the computation time for RSA and RSA DS keys generation of length 1024-bit, 2048-bit, and 3072-bit with ECIES and ECDSA keys of length 160-bit, 224-bit and 256-bit corresponding to symmetric security level of 80-bit, 112-bit, and 128-bit, the simulation shows a faster keys generation for ECIES and ECDSA. ECIES and ECDSA public keys is a point on the curve, and the private keys are generated randomly, this gives them an advantage for generating both keys in a very short time. Figure 8 shows the computation time for encryption and decryption processes for RSA-1024, RSA-2048, and RSA-3072 keys and figure 9 shows encryption and decryption processes for ECIES-160, ECIES-224, and ECIES-256 keys, the simulation shows relatively faster encryption and decryption for ECIES, because ECIES relying on the hardness of the discrete logarithm problem in elliptic curve groups. It consumes low computation time for encryption and decryption and uses small key size.



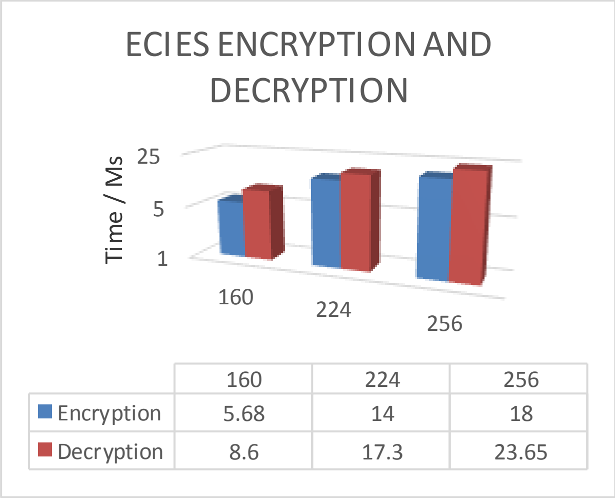
	80 bit	112 bit	128 bit
🟦 RSA and RSA DS	436.87	3755.04	17378.4
🟥 ECIES and ECDSA	21.93	24.22	24.79

Figure 7 RSA and ECIES Keys generation



	1024	2048	3072
🟦 Encryption	2.68	4.2	5.12
🟥 Decryption	10.11	61.32	196.2

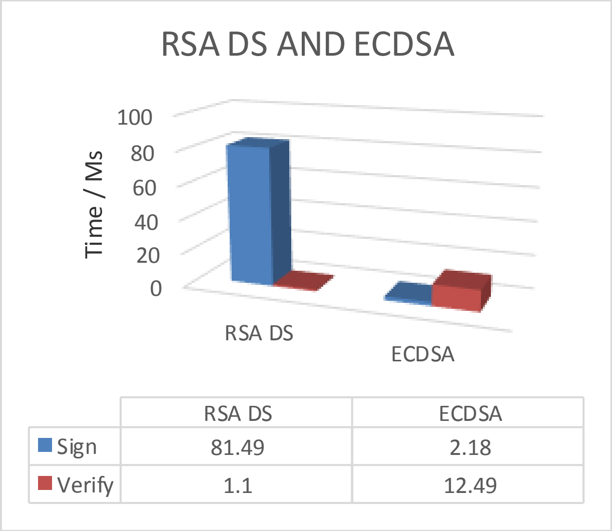
Figure 8 RSA encryption and decryption process.



	160	224	256
🟦 Encryption	5.68	14	18
🟥 Decryption	8.6	17.3	23.65

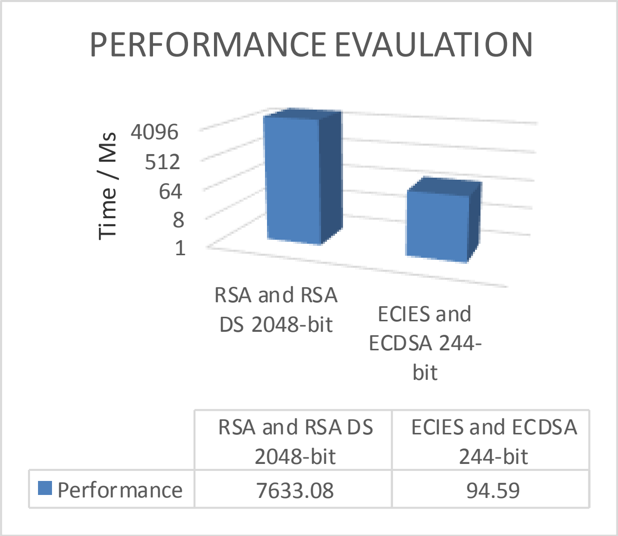
Figure 9 ECIES encryption and decryption process.

Figure 10 shows the average time for signing and verifying processes between RSA DS and ECDSA using the previous keys length. And finally figure 11 and 12 shows the performance evaluation for recommended security keys life time of keys lengths for ECIES and ECDSA of 224-bit, and 256-bit with RSA and RSA DS of 2048-bit, and 3072-bit.



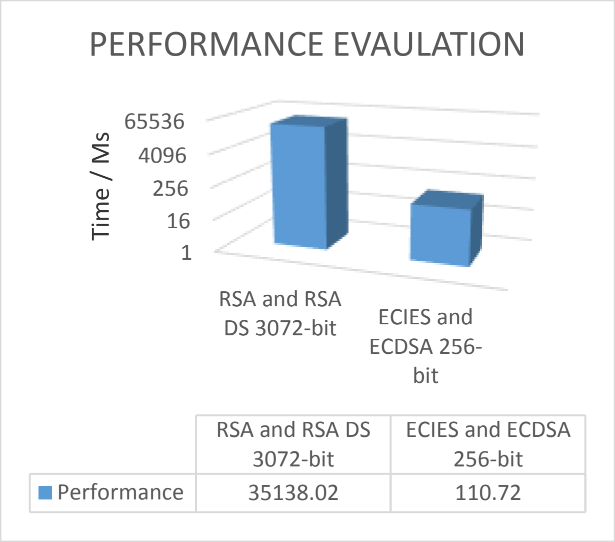
	RSA DS	ECDSA
🟦 Sign	81.49	2.18
🟥 Verify	1.1	12.49

Figure 10 Average time for signing and verifying between RSA DS and ECDSA.



	RSA and RSA DS 2048-bit	ECIES and ECDSA 244-bit
🟦 Performance	7633.08	94.59

Figure 11 Performance evaluation 1.



	RSA and RSA DS 3072-bit	ECIES and ECDSA 256-bit
🟦 Performance	35138.02	110.72

Figure 12 Performance evaluation 2.

VI. Conclusion

This paper shows the possibility of establishing mobile payment system based on Elliptic Curve Cryptography. The security issue of the system depends on the hardness of elliptic curve discrete logarithm problem. Elliptic curve cryptography is actually adopted to its efficiency and requires small key size comparing to RSA cryptosystem with the same security level. This system is considered as an alternative method to displace the current traditional payment systems, in order to insure the security and confidentiality issues. As well as, the propose system needs minimum requirements such as; mobile phone, mobile operator, and market server.

doi:10.15849/icit.2015.0055 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

REFERENCES

[1] L. Bailly and B. Van der Lande, "Breakthroughs in the european mobile payment market," Atos Oringin, 2007.

[2] H. Wilox, "Checkout the mobile payment opportunity," Juiper research, 2007.

[3] R. Markan and K. Gurvinder, "Literature Survey on Elliptic Curve Encryption Techniques," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 9, pp. 906-909, 2013.

[4] M. A. Alia and A. B. Samsudin, "New key exchange protocol based on Mandelbrot and julia fractal sets," International Journal of Computer Science and Network Security, vol. 3, no. 9, pp. 906-909, 2007.

[5] R. A. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.

[6] NaQi, Wei Wei, J. Zhang, W. Wei , J. Zhao, J. Li, P. Shen, X. Yin, X. Xiao and J. Hu , "Analysis and Research of the RSA algorithm," Asian network for scientific Information, vol. 12, no. 9, pp. 1818-1824, 2013.

[7] N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, vol. 48, no. 177, p. 203–209, 1987.

[8] V. Miller , "Use of elliptic curves in Cryptography," Springer-Verlag, vol. CRYPT0 '85, no. LNCS 218, pp. 417-426, 1986.

[9] D. S. Kumar, C. Suneetha and A. ChandrasekhAR , "Encryption of data using elliptic curve over finite fields," International Journal of Distributed and Parallel systems, vol. 3, no. 1, pp. 301-308, 2012.

[10] V. G. Martinez, L. H. Encinas and C. San, "A survey of the elliptic curve inegrated encryption scheme," Journal of Computing Science and Engineering, vol. 2, no. 2, pp. 7-13, 2010.

[11] M. Abdalla, M. Bellare and P. Rogaway, "DHAES: An encryption scheme based on the Diffie-Hellman problem," submission to IEEE http://grouper.ieee.org/groups/1363/P1363a/contributions/dhaes.pdf , 1998.

[12] M. Abdalla, M. Bellare and P. Rogaway, "DHIES: An encryption scheme based on the Diffie Hellman problem," unpublished. http://www.cs.ucdavis.edu/~rogaway/papers/dhies.pdf , 2001.

[13] Girish and Phaneendra , "Identity-Based Cryptography and Comparison with traditional Public key Encryption: A Survey," International Journal of Computer Science and Information Technologies, vol. 5, no. 4, pp. 5521-5525, 2014.

[14] B. Williams, "Will End to End Encryption Save Us All," Brandonwillams Secure Business Growth, 2010.

[15] Infobip, "USSD Interactive Services," 2015. [Online]. Available: http://www.infobip.com/services/ussd/. [Accessed 10 2 2015].

[16] Tekutiev, "USSD Interactive Services," 2013. [Online]. Available: http://eyeline.mobi/blog/author/nikita-tekutiev/. [Accessed 12 2 2015].

[17] S. Karnouskos, A. Vilmos and A. Ram, "SeMoPS: A Global Secure Mobile Payment Service," 2005.

[18] Mashable, 2014. [Online]. Available: http://mashable.com/category/google-wallet/ . [Accessed 12 2 2015].

[19] V. Security, "Infinite Peripherals Partners with Voltage Security to Enhance Mobile Payment Data Protection," 2015. [Online]. Available: http://www.voltage.com/company/news/press-room/pr140210-infinite-peripherals-partners-with-voltage-security-to-enhance-mobile-payment-data-protection/. [Accessed 25 1 2015].

[20] O. Santolalla, "Mobile payment as key factor for mobile commerce success," Helsinki University of Technology, 2008.

[21] VeriFone, "Verishield Total Protect," 2014.

[22] A. Tohari, "Elliptic Curve Cryptography for Securing Payment Sysmtem," in ICTS, Bali, 2013.

[23] A. Hnaif and M. Alia, "Mobile payment method based on public-key cryptography," International Journal of computer networks & communications, vol. 7, no. 2, pp. 81-92, 2015.

[24] FlexiProvider. [Online]. Available: https://www.flexiprovider.de/. [Accessed 23 10 2014].