Modified SET Protocol for Mobile Payment: An Empirical Analysis

Sabrina M. Shedid[†], Magdy El-Hennawy[††] and Mohamed Kouta[†††]

British University in Egypt, Higher Institute for Computers and Information Technology, Arab Academy for Science, Technology and Maritime

Summary
The wide spread of using handheld devices offers an opportunity for mobile devices to be used as a universal payment method. However, some issues impede the widespread acceptance of mobile payment; for example: privacy protection, limited capability of mobile devices, and limited bandwidth of wireless networks. In ecommerce payment, Secure Socket Layer (SSL) protocol has been used to establish a secure channel between customers and merchants to secure the payment and the order information. SSL has some disadvantages regarding customer privacy that the customer payment information is revealed to the merchant. Secure Electronic Transaction (SET) has resolved SSL protocol disadvantages by dividing order message into: 1) order information which is revealed to merchant M, 2) payment information which is revealed to Payment Gateway (PG). Both SSL and SET assume the existence of Public Key Infrastructure (PKI) where extensive computations are carried out. In mobile payment, the same protocols of ecommerce payment are used but their application is limited due to heavy computations over wireless and GSM networks. A Modified Secure Electronic Transaction (MSET) protocol is proposed to minimize the extensive computations of SET protocol through replacing time consuming public key encryption and decryption algorithms by symmetric key cryptography.

Key words:
M-Commerce, Wireless Security, Cryptography

1. Introduction

We define m-commerce as using a mobile device for business transactions performed over a mobile telecommunication network, possibly involving the transfer of monetary values [10, 19]. M-commerce is not just about using mobile phones as end user devices. The following list gives an overview of different kinds of mobile devices: Mobile phone, Personal Digital Assistant (PDA), Smart phones, and Laptops. Each mobile device has certain characteristics that influence its usability, such as size and color of display, input device, availability of keyboard and mouse, memory and CPU processing power, network connectivity, bandwidth capacity, supported operating systems, and availability of internal smart card reader (e.g. a SIM card in mobile phones). Depending on these factors, services that end user can receive differ considerably. Moreover, depending on the network technology used for transmission, the variation in bandwidth capacity also influences the kind of services that the end user is able to receive. Using a mobile phone as a universal payment instrument requires considerable reduction in the computational requirements of the existing payment protocol standards. A secure mobile payment protocol based on Simple Initiation Protocol (SIP) and public key cryptography was proposed [24]. A light weight secure electronic protocol was proposed [13]. The computational reduction is done through what is called message linkage [14, 11]. The authors claimed about 50% reduction in computation, and 80% reduction in communication overhead. A protocol that provides identity protection for the wallet is presented [13]. The proposed protocol incorporates the Mobile Network Operator (MNO). A simple secure M-Commerce Protocol was proposed [12]. The protocol utilizes the Transport Layer Security Protocol (TLS) and the Wireless TLS (WTLS) in lower layers to reduce the number of required signature generations within the protocol. In our paper we will proposes a Modified SET (MSET) protocol with the following characteristics: ***First***, the MSET is shared by the same parties as SET Wallet (W), Merchant (M), and the Payment Gateway (PG) which makes the MSET an acceptable universal payment instrument. ***Second***, the MSET uses in the set up (initialization, registration) phase the public key infrastructure for symmetric key exchange [22, 20]. Diffi-Helman key exchange protocol can be used, but the Diffi-Helman key exchange protocol [6] does not support a proof of identity [2, 18] for both parties of the protocol. ***Third***, according the analysis shown in [16, 15, 17], comparing the efficiency of the symmetric key encryption and decryption algorithms with asymmetric key encryption and decryption algorithms, MSET speed up ratio exceeds all existing protocol at all parties. In this paper, we will focus on the reduction of the computation at W which enables the Mobile phone to be used as a payment device. ***Fourth***, MSET ensures the same security characteristics as SET. MSET establishes secure channel for communication between parties, achieving privacy, authenticity, non repudiation, and message integrity. MSET keeps the privacy of the payment information of W away from M and keeps the privacy of the order

Manuscript received July 5, 2010
Manuscript revised July 20, 2010

information of W away from the PG. MSET enables the PG to verify that M does not alter the Purchase Request (PReq) message during its processing at M. The only assumption we make is to assume that the PG is a Trusted Third Party (TTP).The assumption that PG is trusted is a fair assumption. If the PG and M are cheating together, they can create a Purchase Request message as if it was originated from W and charges W for the value of the transaction. The paper is organized as the following:-In section II, The SET protocol is explained. In section III, MSET protocol is proposed. In section IV, we analyze the proposed protocol. Finally, in section V, we come to a conclusion and discuss some issues for further work.

2. The SET Protocol

The SET protocol involves three parties: Customer digital wallet W, Merchant M, and payment gateway G. The MSET and SET notations are shown in table 1.

Table 1: SET & MSET Notations & Acronyms

Notation	Meaning
W	Mobile digital wallet
M	Merchant
PG	Payment Gateway
PinitReq	Purchase initiate Request message
PinitRes	Purchase initiate Response message
PreReq	Purchase Request message
AuthReq	Authorization Request message
AuthRes	Authorization Response message
Pres	Purchase Response message
PReq	Purchase Response message
H	One way hash function
ES	Encryption function using symmetric key
DS	Decryption function using symmetric key
EA	Encryption function using asymmetric key
DA	Decryption function using asymmetric key
Pubw	Wallet Public Key
Privw	Wallet Private Key
Pubm	Merchant Public Key
Privm	Merchant Private Key
Pubpg	PG Public Key
Privpg	PG Private Key
$S_{(w,m)}$	Symmetric Key between wallet & merchant
$S_{(w,pg)}$	Symmetric Key between the wallet and the PG
$S_{(pg,m)}$	Symmetric Key between the PG and the merchant
$S_{(w,m)}$	Symmetric Key between wallet & merchant
OM	Order information Message
PM	Payment information Message
SK	Session Key
DC_w	Wallet Digital Certificate
DC_m	Merchant Digital Certificate
DC_{pg}	PG Digital Certificate

PinitReq: W → M

The wallet application performs the following steps:

1. H(PinitReq) to get the of the hash of the message.
2. $EA_{Privw}(H(PinitReq))$ to sign the message.
3. $ES_{SK}(PinitReq||EA_{Privw}(H(PinitReq))||DC_w$ to get the ciphered message using symmetric session key SK.
4. $EA_{Pubm}(SK)$ to get the message envelope.
5. W sends both the ciphered message and the envelope to M.

Purchase initiate request from the e-wallet to the Merchant. M is signed by W using public key cryptography like RSA. Both the message and the wallet signature are encrypted by randomly generated session key SK using symmetric key encryption like DES or AES to get the ciphered message. The Digital envelope is created by encrypting the session key SK by the Merchant public key to achieve message privacy. Both the Encrypted Message and the digital envelope are sent to M. The PinitReq message contains among other things the credit card brand name (not credit card number), bank identification number (the issuer bank), a challenging string to be used by M in his response to W, and Wallet digital certificate which contains the wallet public key. In this message, the public key encryption is applied twice, and the symmetric key encryption is applied once. When M receives the message and the envelope, the following steps are performed:

1. $DA_{Privm}(EA_{Pubm}(SK))$ to get SK.
2. Using SK $DS_{SK}(ES_{SK}(PinitReq||EA_{Privw}(H(PinitReq))||DC_w))$ to get $PinitReq||EA_{Privw}(H(PinitReq))||DC_w$.
3. Get the hash of the received message H(PinitReq).
4. Getting the Wallet public key Pubw from DC_w and apply $DA_{Pubw}(EA_{Privw}(H(PinitReq)))$.
5. Compare between the results of step 3 and step 4 to verify the integrity of the message, non repudiation and the authenticity of W.

M opens the envelope of PinitReq message using his private key to get the session key SK then the whole message and the signature are decrypted. The hash of the received message is compared with the hash of the originally signed message after decrypting it using the wallet public key.

PinitRes M → W

M send purchase initiate request message (PinitReq) to W. The PinitReq contains a unique transaction identification number, challenging string, and merchant digital certificate. For W to read and verify PinitRes, the public key decryption algorithm is applied twice, and the symmetric key decryption is applied once.

PReq W → M

Purchase request (PReq) is a doubly signed message. The wallet partitions the message into two sub messages. The first one contains the Order information Message OM where an envelope is created using M public key such that it can only read by M. The second message contains only the Payment information Message PM and a digital envelope is created using PG public key such that it can only read by PG. Each message is signed by W. The hash of the first message, the hash of the second message are concatenated together, hashed again and signed by W to ensure the integrity of the whole message as shown in Fig. 1. The following steps are performed as follows:

1. H(OM).
2. $EA_{Privw}(H(OM))$.
3. $ES_{SK1}(OM||EA_{Privw}(H(OM))||DC_w)$ For randomly generated session key SK1.
4. $EA_{Pubm}(SK1)$ to get the M envelope.
5. H(PM).
6. $EA_{Privw}(H(PM))$.
7. $ES_{SK2}(PM||EA_{Privw}(H(PM))||DC_w)$ for randomly generated session key SK2.
8. $EA_{Pubpg}(SK2)$ to get the PG envelope.
9. H(H(OM)||H(PM)).
10. $EA_{Privw}(H(H(OM)||H(PM)))$ to get the doubly signed message.

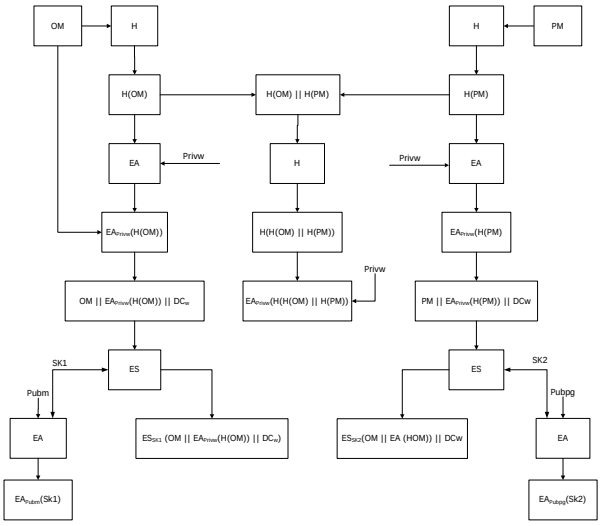

Fig. 1 PReq Doubly Signed Message by W in SET

In this case, the public key encryption is applied three times, and the symmetric key encryption is applied twice. For M to verify the PReq message as shown in Fig. 2, the following steps are performed:

1. $DA_{Privm}(EA_{Pubm}(SK1))$ to get SK1.
2. Using SK1 $DS_{SK1}(ES_{SK1}(OM||EA_{Privw}(H(OM))||\ DC_w))$ to get $(OM||\ EA_{Privw}(H(OM))||DC_w)$.
3. Get the hash of the received message H(OM).
4. Get the Wallet public key Pubw from DC_w and perform $DA_{Pubw}(EA_{Privw}(H(PM)))$ to get H(PM).
5. H(H(OM)||H(PM)).
6. $DA_{Pubw}(EA_{Privw}(H(H(OM)||H(PM))))$ to get H(H(OM)||H(PM)).
7. Compare between the results of step 5 and step 6 to verify the integrity of the both messages.

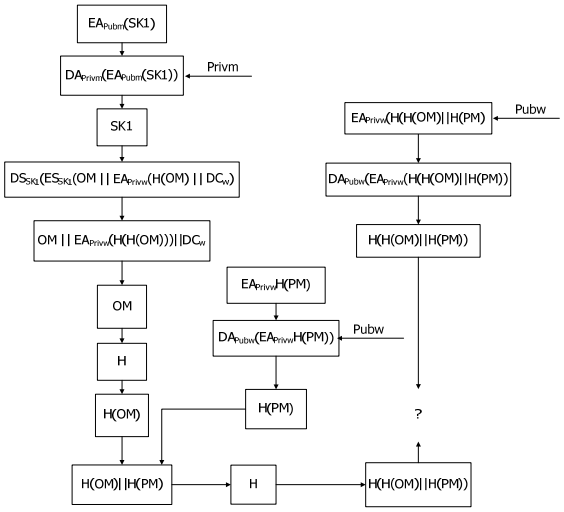

Fig. 2 M verifies the doubly PReq Message in SET

AuthReq M → PG

Here M request authorization from PG. M sends the second message PM from PReq with envelope using the PG public key and adds to it, among other things additional information related to his financial institution and his digital certificate to the PG. The settlement (the actual money transfer) from the issuer bank (Wallet Bank) to the acquirer bank (Merchant bank) is done through Electronic Fund Transfer Network (EFTN) and PG gets notified.

AuthRes PG → M

After PG checks open – to - buy availability, the PG sends authorization response to M.

PRes M → W

After M gets the authorization from PG, M sends Purchase response message to W and this completes the protocol. For W to read and verify the PRes message, the public key decryption algorithm is applied twice, and the symmetric key decryption is applied once.

3. The Proposed MSET Protocol

MSET protocol replaces public key encryption by symmetric key encryption. This involves two steps. Set up step (registration step) and the transaction step.

Set up step (Applied once)

Here the three parties W, M, and PG exchange their digital certificates. Three symmetric keys are generated $S_{(w,m)}$, and $S_{(m,pg)}$, and $S_{(w,pg)}$ The three keys are exchanged as follows:

1. M generates $S_{(w,m)}$ encrypts it with W public key and sends it to W where revealed to W using W private key.
2. PG generates $S_{(m,pg)}$ and encrypts it with M public, sends it to M where revealed to M using M private key.
3. PG generates $S_{(w,pg)}$ and encrypts it with W public, sends it to W where revealed to W using W private key.

We assume that each party has a lookup table that includes the digital certificate of all the other parties associated with their symmetric keys. The protocol shown in [22] can be used.

Transaction step (Applied every Transaction)

This involves the same six steps as follows:

PinitReq: W → M

The concatenation of PinitReq, the hash of PinitReq, and DC_w is ciphered by $S_{(w,m)}$ using symmetric key encryption. Message creation needs application of symmetric key encryption once. The following steps are performed as follows:

1. H(PinitReq) to get hash of the message.
2. $ES_{S(w,m)}(PinitReq||H(PinitReq)||DC_w)$ to get the ciphered message using the symmetric key $S_{(w,m)}$.
3. W sends the ciphered message to M.

To form the message, symmetric key encryption is performed once using same key $S_{(w,m)}$. The received message is decrypted by M using $S_{(w,m)}$. The hash of the received message is compared to the hash of the sent message to ensure message integrity. Since $S_{(w,m)}$ is only known to W and M, this ensures privacy, authenticity, and non repudiation. The following steps are performed by M:

1. $DS_{S(w,m)}(PinitReq||H(PinitReq)||DC_w)$ to get
2. $PinitReq||H(PinitReq)||DC_w$.
3. Compute the Hash of the received message H(PinitReq).
4. Compare the received hash in step 2 with the computed hash in step 3 to verify the integrity of the message.

PinitRes M → W

The same analysis applies meaning message creation needs application of symmetric key encryption once. To read the message, symmetric key decryption is performed once using same key $S_{(w,m)}$.

PReq W → M

Here the message that contains order information and its hash is ciphered with $S_{(w,m)}$ where the message that contains payment information and its hash is ciphered using $S_{(w,pg)}$. The double signing is done using $S_{(w,m)}$. In SET, the signature of W on OM and PM and the double signature are done using W private key where it can be verified using W public key which is known to M and W. In the proposed MSET symmetric key cryptography are used where $S_{(w,m)}$ is not known to PG and $S_{(w,pg)}$ is not known to M, for M to verify OM and the whole PReq message and for PG to verify PM message and the integrity of the whole message, each signature is carried twice: once using $S_{(w,m)}$ and another time using $S_{(w,pg)}$ as shown in Fig 3. The following steps are performed as follows:

1. H(OM).
2. $ES_{S(w,m)}H(OM)$ to get the sign on the order message to be verified by the Merchant.
3. $ES_{S(w,pg)}H(OM)$ to get the sign on the order message to be verified by the Payment Gateway to ensure that the Merchant does not change the order message.
4. $ES_{S(w,m)}(OM)$ to get the cipher message, this message can be revealed by the Merchant only preserving the wallet privacy.
5. H(PM).
6. $ES_{S(w,m)}H(PM)$
7. $ES_{S(w,pg)}H(PM)$
8. $ES_{S(w,pg)}(PM)$.
9. H(H(OM)||H(PM)).
10. $ES_{S(w,m)}(H(H(OM)||H(PM)))$.
11. $ES_{S(w,pg)}(H(H(OM)||H(PM)))$.

As shown in Fig 4, When M receives the message; the following steps are performed as follows:

1. $DS_{S(w,m)}(ES_{S(w,m)}(OM))$ to get OM.
2. $DS_{S(w,m)}(ES_{S(w,m)}(H(PM)))$ to get H(PM).
3. $DS_{S(w,m)}(ES_{S(w,m)}(H(H(OM)||H(PM))))$ to get H(H(OM)||H(PM)).
4. H(OM) from step 1.
5. Get the hashing of the Concatenation of steps 4 and 2.
6. Compare the result of steps 5 and 3.

AuthReq M → PG

After M receives the OM message, the Merchant compute the following:

1. H(OM)
2. $ES_{S(m,pg)}H(OM)$ and sends $ES_{S(m,pg)}H(OM)$ and $ES_{S(w,pg)}H(OM)$ such that the PG can verify that the Merchant does not alter the OM message during the processing of the PReq message at the Merchant site.

The PG can verify the PM message and the double signature on the PReq message using the same steps as the Merchant did as previously mentioned as shown in Fig 5.

AuthRes PG → M

Message creation needs the application of symmetric key encryption once $S_{(m,pg)}$. To read the message, symmetric key decryption is performed once using same key $S_{(m,pg)}$.

PRes M → W

Message creation needs the application of symmetric key encryption once $S_{(w,m)}$. To read the message, symmetric key decryption is performed once using same key $S_{(w,m)}$.

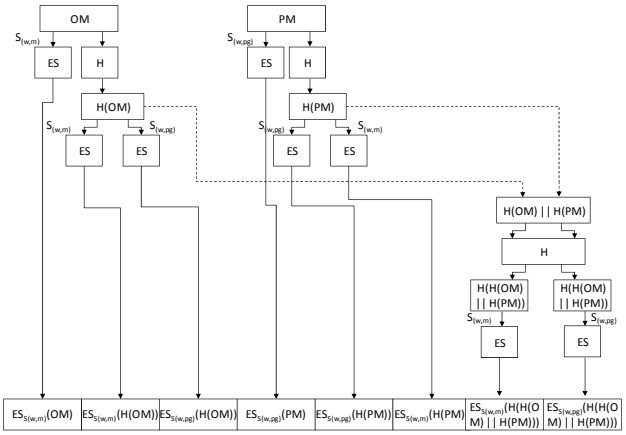

Fig. Fig. 3 PReq Message (MSET)

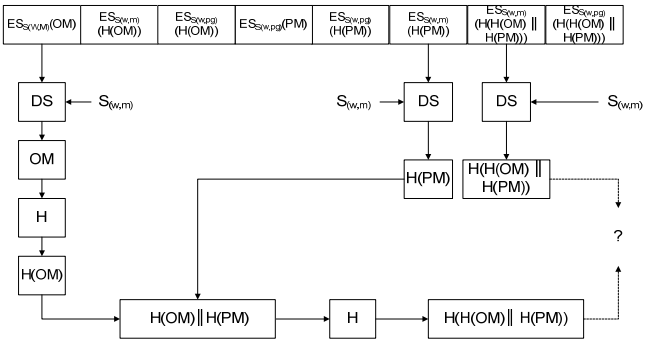

Fig. 4 M verifies Preq Message (MSET)

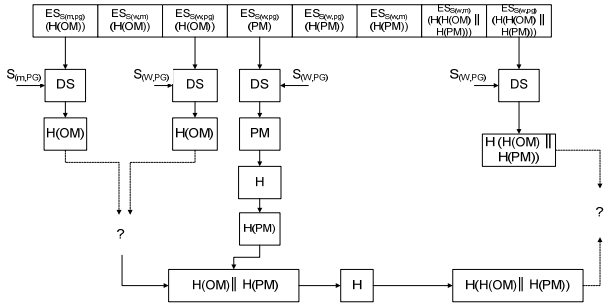

Fig. 5 PG verifies that the Merchant does not change the content of PRes (MSET)

4. The Analysis of the Proposed Protocol

Based upon [16], the proposed MSET protocol dramatically decreased the computational time to perform the wallet operations and maintains the same level of security. Accordingly, it is possible to use the new protocol in mobile devices as universal payment standard. Table II shows the operations of SET and MSET at the wallet. Table III summarizes the operations performed on the wallet side in both the SET and the modified SET.

Table 2: SET Vs. MSET

STEP	SET	MSET
PinitReq	2 Asymmetric key encryptions & 1 Symmetric key encryption	1 Symmetric key encryption
PinitRes	2 Asymmetric key decryptions & 1 Symmetric key decryption	1 Symmetric key decryption
PReq	5 Asymmetric key encryptions & 2 Symmetric key encryptions	8 Symmetric key encryptions
PRes	2 Asymmetric key decryptions 1 Symmetric key decryption	1 Symmetric key decryption

Table 3: Number of Symmetric and Asymmetric Keys for SET & MSET

SET	MSET
7 Asymmetric key encrypt.	9 Symmetric key encryption
3 Symmetric key encrypt.	2 Symmetric key decryption
4 Asymmetric key decrypt.	
2 Symmetric key decrypt.	

Table IV was quoted from [21]. The values appeared on the table were based on experimentations carried out on compact IPAQ – H3630 Pocket PC.

Table 4. Timing measurements of low-level cryptographic primitives

Operation	Time	Iterations
DES	7.354 sec. (7,354 ms)	100,000 encryptions & 100,000 decrypt.
SHA	19.111 sec. (19,111 ms)	100,000
1,024 bits RSA signing	782.593 sec. (782,593 ms)	10,000
1,024 bits RSA verification	50.125 sec. (50,125 ms)	10,000
2,048 bits RSA signing	4,972.798 sec. (4,972,798 ms)	10,000
2,048 bits RSA verification	156.006 sec. (156,006 ms)	10,000

Reference to table IV, the one iteration computational time for DES encryption or decryption with 56 bit key: (7,354/100,000 = 0.07345 ms), RSA Signing with 1,024 bit key: (782,593/10,000 = 78.2593 ms), and RSA Verification with 1,024 bit key: (50,125/10,000 = 5.0125 ms).

Reference to table III, SET computational time = (1,024 bits RSA signing computational time for one iteration * SET Asymmetric key encryptions) + (1,024 bits RSA verification computational time for one time * SET Asymmetric key decryptions) + (DES computational time for one iteration * (SET Symmetric key encryptions + SET Symmetric key decryptions)) and MSET computational time = (DES computational time for one iteration * (MSET Symmetric key encryption + MSET Symmetric key decryption)). Therefore SET computation time = (78.2593*7) + (5.0125*4) + (0.07354*(3+2)) = 568.2328 ms and MSET computational time = (0.07354*(9+2)) = 0.80894 ms.

The speedup ratio = (SET computational time/MSET computational time) = (568.2328/0.80894) = 702.4412 and the computational reduction percentage = (((SET computational time - MSET computational time)/(SET computational time))*100) = ((568.2328-0.80894)/(568.2328)*100) = 99.85%. These calculations reflect the speed up ratio and the percentage of reduction as shown in fig 6. The values may vary depending upon the asymmetric algorithms used (RSA, Elliptic Curve), the key length (512, 1024, 2048), the symmetric algorithms used (DES, Triple DES, AES), the key length (56, 128, 192), and the computation domain (Integer, Galois field (2^n)).

5. Conclusion & Future Work

A modified version of SET protocol was proposed. It did not only preserve the main security features of traditional approaches but also tackled the computational complexity compared to SET. As the computational complexity decreased, the mobile battery power consumption is reduced as well [5]. The following suggestions are presented for the future work: ***First***, securing the use of mobile wallet to be used only by authenticated mobile owners. Biometric recognition may be deployed such as finger print, iris, voice, or face recognition. ***Second***, for the network reliability: uncompleted transaction needs a strong recovery mechanism to ensure the atomic property of the transaction. ***Thirdly***, for the customer privacy problem: the current algorithms allow PG to trace the customer behavior. ***Fourthly***, the mobile wallet needs to store digital money with the same adaptability as traditional paper money. Digital Money can be spent offline and it can be transferred, exchanged where divisibility and traceability are guaranteed.

REFERENCES

[1] A. Fourati, H. Ben Ayed, F. Kamoun, and A. Benzekri, "A SET Based Approach to Secure the Payment in Mobile Commerce," 27th Annual IEEE Conference on Local Computer Networks (LCN'02), Nov 2002, pp. 136-140.

[2] A. K. Ghosh, and T. M. Swaminatha, "Software security and privacy risks in mobile e-commerce," Communications of the ACM, vol. 44, pp. 51–57, February 2001.

[3] C. Brookson, "GSM security: A description of the reasons for security and the techniques," in Proc. IEE Colloquium on Security and Cryptography Applications to Radio Systems, pp. 2/1–2/4, June 1994.

[4] Biswas S, and Neogy S, A Mobility – Based Check Pointing Protocol for Mobile Computing System, IJCSIT VOL 2, N0 1, February 2010.

[5] D. Shah, and S. Zhong, "Benchmarking Security Computation on Wireless Devices," the 3rd International Conference on Informatics and Technology, Aug. 2009.

[6] Diffie W., and Hellman M, New Direction in Cryptography, IEEE Transaction on Information Theory, IT-22(6):644-654, 1976.

[7] Fun T. S., Beng L. Y., Roslan R., and Habeeb H. S.Privacy in New Mobile Protocol, World Academy of Science, Engineering, and Technology 47, 2008.

[8] G. Apostolopoulos, V. Peris, P. Pradhan, and D. Saha, "Securing electronic commerce: Reducing SSL overhead," in IEEE Network, pp. 8–16, July 2000.

[9] G. Lawton, "Biometrics: A new era in security," IEEE Computer, vol. 31, pp. 16–18, Aug. 1998.

[10] H. Beadle, R. Gonzalez, R. Safavi-Naini, and S. Bakhtiari, "A Review of Internet Payments Schemes," In Proceedings of the Australian Telecommunication Networks and Applications Conference (ATNAC'96), Melbourne, Australia, December 1996, pp.486-94.

[11] H. Wang, and E. Kranakis, "Secure Wireless Payment Protocol," International Conference on Wireless Networks 2003. pp. 576-582.

[12] Haddad E., and King B, A Simple Secure M Commerce Protocol SSMCP, IJCSNS International Journal of Computing and Network Security, VOL 7, No 3, March 2007.

[13] Hanoka G., Zheng Y., and Imai H. Improving the Secure Transaction Protocol by Using Sign crypt, IEICE Transactions Fundamentals VOl E84A, No 8, August 2001.

[14] J. Hall, S. Killbank, M. Barbeau, and E. Kranakis, "WPP: A Secure PaymentProtocol for Supporting Credit- and Debit-Card," International Conference on Telecommunications, Romania, Bucharest, June 4-7, 2001.

[15] J. Liu, et al., A System Model and Protocol for Mobile Payment, in proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE' 05), 2005.
[16] Lamercht C., Van Moorsel A., Tomlinson p., and Thomas N, Investigating the Efficiency of Cryptographic Algorithms in Online Transactions , International Journal of Simulation VOL 7, No 2.
[17] M. Hassinen, An Open, PKI-Based Mobile Payment System, in proceedings of Emerging Trends in Information and Communication Security, Freiburg, 2006.
[18] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in Proc. ACM Int. Conf. Mobile Computing and Networking, pp. 180–189, July 2001.
[19] N. Kreyer, K. Pousttchi and Klaus Turowski, "Standardized Payment Procedures as Key Enabling Factor for Mobile Commerce", in proceedings of E-Commerce and Web Technologies: Third International Conference, 2002.
[20] N. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayana, "Optimizing Public-Key Encryption for Wireless Clients," in Proc. IEEE Int. Conf. Communications, pp. 1050–1056, May 2002.
[21] P. Argyroudis, R. Verma, H. Tewari, and D. O'Mahony, "Performance analysis of cryptographic protocols on handheld devices," The Third IEEE International Symposium Network Computing and Applications, 2004. (NCA 2004), Cambridge, MA, Aug 30.
[22] Parnerkar A, Guster D, and Herald J, Secret Key Distribution Protocol Using Public Key Cryptography, JCSC VOL 19, No 1, October 2003.
[23] S. Friis-Hansen, and B. Stavenow, "Secure Electronic Transactions-The Mobile Phone Continues," Ericsson Review, no. 4, 2001. http://www.ericsson.com/about/publications/review/200104/files/2001041.Nake
[24] Zhang G, Cheng F, and Meinel C, Towards Secure Mobile Payment Based on SIP 15th Annual IEEE International Conference and Workshop on the Engineering Based System, 2008.

Sabrina Shedid received the B.S. degree in Computers and Information Systems from October 6 University in 2003. During 2007-2010, she is preparing the M.S. degree in Computers and Information Systems from the Arab Academy for Science, Technology and Maritime Transport. All of her researches are concerned with M-Commerce and Wireless Security.

Mohamed Kouta received the B.S. degree in Electrical Engineering from Military Technical College in 1972. He received the M.S. and Ph.D. degrees in computer science from Jons Hopkins University and Clarkson University in 1982 and 1985, respectively. He is the Chairman of Business information system(BIS) Department (Cairo Branch),college of management and technology, Arab academy for science and technology (AAST) and the Vice Dean for Education

Magdy El-Hennawy is lecturer in the Higher Institute of Computer Science & Information Technology, El-Shorouk Academy, and in the same time working as a Project manager of the family card system in Ministry of State for Administrative Development, MSAD. Before and since 1978 working as manager of SW development and maintenance center specialized in mission critical SW systems development. I have working in that place as deputy manager, chief of the system engineering team, member in a team that manages the system analysis/design and implementation of SW systems, at the same time some researches and courses teaching are done, and member in a project organization, chief of a development team for the analysis, design, and implementation