2012 Third International Conference on Computer and Communication Technology

Enhanced Mobile SET Protocol with Formal Verification

Shaik Shakeel Ahamad[1, 2], V.N.Sastry[1] and Siba K. Udgata[2]
[1]Institute for Development and Research in Banking Technology (IDRBT), Castle Hills, Masab Tank, Hyderabad-57, India
and
[2]Department Computers and Information Sciences, University Of Hyderabad, Hyderabad-46, India
ahamadss786@gmail.com, vnsastry@idrbt.ac.in and udgatacs@uohyd.ernet.in

***Abstract*— In this paper we propose an Enhanced Mobile SET (EMSET) protocol with formal verification using Mobile Agent technology and Digital Signature with Message Recovery based on ECDSA mechanism. Mobile Agent technology and Digital Signature with Message Recovery (DSMR) based on ECDSA mechanism provides in proposing EMSET protocol in Mobile Networks. Mobile Agent technology has many benefits such as bandwidth conservation, reduction of latency, reduction of completion time, Asynchronous (disconnected) communications. Digital Signature with Message Recovery based on ECDSA eliminates the need of adopting PKI cryptosystems. Our proposed protocol EMSET ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these our proposed protocol withstands Replay, Man in the Middle and Impersonation attacks. The security properties of the proposed protocol have been verified using Scyther Tool and presented with results.**

Keywords- EMSET, Mobile agents, Digital Signature with Message Recovery (DSMR) Money laundering, Double Spending, Overspending and Scyther Tool.

I. INTRODUCTION

Mastercard and VISA have introduced SET protocol [1] which is a very popular credit-card payment protocol. SET was successfully implemented on fixed networks but it is not easy to implement it on wireless networks because of the nature of the SET itself and the problems in wireless networks. SET is a complex protocol which is implemented using public-key infrastructure (PKI). Adopting PKI to wireless environment is a non trivial task because of the limitations of the wireless communication environment when implementing wireless PKI. Due to these problems, it is very difficult to apply wired PKI system to the wireless environment [6, 7]. To overcome these limitations, authors of [2] proposed an agent-based SET payment system (SET/A). With SET/A, client is not required to stay connected to the Internet during the whole period of the transaction. An agent containing SET wallet plays the client's role in SET payment session. Thus, the client needs to connect to the Internet for short periods during the entire transaction. However, SET/A is vulnerable to attacks because the agent is required to bring SET wallet with it to perform cryptographic operations at the merchant environment which is considered to be hostile so Non repudiation property is not ensured. Authors of [3] proposed SET/A+ which is a modified version of SET/A in order to solve the problems and limitations of SET/A. SET/A+ is operated in the larger scenario than that of SET/A, in that, it includes the brokering and negotiation phase which naturally requires the capability of agent in SET protocol. Client's Purchase Request is completely generated on the client's mobile device before it is brought with an agent to merchant. SET/A+ solves the problem of key compromise at the merchants site but performing all the cryptographic operations at clients site results in the problem of high computational load for the client. Moreover signature can be abused easily in malicious merchant environment, so Non repudiation property is not ensured. In order to overcome the limitations SET, SET/A and SET/A+ authors of [4] have proposed to employ the combination of proxy –based solution and the agent technology to secure transactions and solve the problems of implementing SET payment in wireless environments. But the solution provided in [4] has to trust a proxy server, the client need to validate certificate of the issuer every time it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client, if digital signatures are generated in the memory of the mobile phone these signatures should not be considered as signatures because they are not generated in a tamper resistant device. Authors of [5] have proposed SETNR/A protocol was proposed to improve the weakness of lacking non-repudiation mechanism from SET and SET/A for credit card-based transactions; on the other hand, agent-based protocol is ideal for complicated payment system. Broker generates a mobile agent for Buyer (i.e. client) which carries encrypted purchase order to Seller (i.e. merchant). A trusted third party (TTP) acts as a lightweight notary for evidence generations. But the solution provided in [5] has to trust a proxy server, the client need to validate certificate of the Broker, merchant every time it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client, if digital signatures are generated in the memory of the mobile phone these signatures should not be considered as signatures because they are not generated in a tamper resistant device. Authors of [12] propose LITESET/A+ which does not ensure non repudiation property. In this paper, we propose a new mobile payment protocol called EMSET based on SET which is able to solve the problems that are presented in [1, 2, 3, 4, 5 & 12]. Protocols proposed in [1, 2, 3, 4, 5 & 12] are not formally verified. So these protocols cannot be considered as secure protocols. We employ the combination of Mobile Agent technology and Digital Signature with

978-0-7695-4872-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCCT.2012.65

Message Recovery (DSMR) mechanism in order to solve the problems of implementing SET payment in wireless environments. UICC is used a secure element for securely keeping client's credentials and for generating digital signatures. UICC is considered as SSCD (Secure Signature Creation Device) so digital signatures generated can be considered as legal signatures. We have verified our proposed EMSET protocol using Scyther Tool which is an automatic push-button tool for the verification and falsification of security protocols.

II. BACKGROUND

A. Mobile Agent Technology

Mobile agents are considered to be an alternative to client-server systems, in particular for mobile commerce where mobile devices and communications have limited computing resource.

B. DSMR (Digital Signature with Messsage Recovery) mechanism

Digital Signature with message recovery using self-certified public keys [8, 9], provides an authenticated encryption scheme that integrates the mechanisms of signature and encryption, which enable only the specified receiver to verify and recover the original message. The authentication of the public key can implicitly be accomplished with signature verification.

C. Proxy Certificates

The proxy certificate basically follows standard certificate format (X.509) with minor change. The major difference is the subject identifier (SID), which is the certificate field, recorded the owner of this certificate. In proxy certificate, its subject identifier is equal to the certificate issuer. A proxy certificate (PC) of a mobile agent is issued and digitally signed by its owner. Beside standard certificate fields, this certificate contains a set of constraints which specifies valid operations that the agent is allowed to perform while using this certificate.

$Cert\_Ver(PC\{C, K_{Ag}[D]\}_{K_c^{-1}})$ is successful if and only if $Sign\_Ver(PC\{C, K_{Ag}[D]\}_{K_c^{-1}}$ is successful namely

$K_c(PC\{C, K_{Ag}[D]\}_{K_c^{-1}})$ = H$(PC\{C, K_{Ag}[D]\}_{K_c^{-1}})$

We use the notation $(PC\{C, K_{Ag}[D]\}_{K_c^{-1}})$ to represent the proxy certificate of the Mobile Agent (Ag) belonging to its owner Client (C) with data 'D'.

D. Trusted Processing Environment (TPE)

A TPE in a mobile agent system provides a safe environment for the execution of any alien program; these include Software-based fault isolation and safe-code interpretation.

E. Contributions made

1) We have proposed an Enhanced Mobile SET (EMSET) Payment Protocol in mobile networks using Mobile Agents and Digital Signature with Message Recovery (DSMR) based on ECDSA.

2) Our protocol is proposed in the UICC of Mobile Device which is considered to be a tamper resistant device so UICC is a Secure Signature Creation Device (SSCD) because the signature processes are performed in the UICC and the private key never leaves the WIM. So non repudiation is ensured in devices where private key is stored in WIM.

3) The transaction flow in our proposed mobile payment protocol (EMSET) is from client to Issuer decreasing the risk of reusing client's information (PI) for the later transactions and issuer is a trusted entity of the client. So client can trust the TPE (Trusted Processing Environment) of the Issuer.

4) In our proposed EMSET client need not register itself with merchant in merchant registration protocol thereby reducing the consumption of resources.

5) Our proposed EMSET Protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering.

6) In addition to these EMSET Protocol withstands Replay, Man in the Middle and Impersonation attacks.

7) Our proposed EMSET Protocol has been verified successfully using Scyther tool, the results using this tool are given in Section V.

III. OUR PROPOSED ENHANCED MOBILE SET (EMSET) PROTOCOL

A. Assumption

We have modified system initialization phase from [9] i.e. when a client C wants to join the system, he has to prove his credentials and requests for anonymous identity $anonid_c$ then CA allocates anonymous identity to the client so instead of C, CA issues $anonid_c$ as client's identity thereby achieving anonymity.

B. EMSET Protocol

Entities involved in EMSET protocol: Client (C), Merchant (M), Issuer (I), Acquirer (A) and Payment Gateway (PG). C & M trust their banks. Payment Gateway (PG) acts as an arbiter; Issuer (I) authorizes Payment Information (PI) and Merchant (M) authorizes Order Information (OI). Client (C) initiates the payment transaction for purchasing goods from the merchant (M). There are four phases in EMSET protocol they are 1) Registration Phase 2) Negotiation Phase 3) Payment Phase and 4) Deposit Phase.

1) Registration Phase

Step 1: Client delegates the validation of Issuer's (I) certificate to OCSP (Online Certificate Status Protocol).
Step 2: If the response from OCSP is positive go to Step 3
Step 3: Client sends its certificate to the Issuer (I)
Step 4: Issuer (I) validates client's certificate.
Step 5: Client downloads Mobile Payment Application from Issuer (I) and sends the following message to the Issuer (I) $C \rightarrow I: \{M1, SIG_I^C(M1)\}_{K_I}, Cert_c$
Where $M1 = \{PI, phno, NRP, T_c, N_c$ UICC initiates the process to negotiate shared symmetric key with the Bank and sends $SIG_I^C(M1)\}_{K_I}, Cert_c$ where $M1 = \{PI, phno, NRP, T_c, N_c$. Issuer (I) decrypts the received message from UICC using his private key and checks the authenticity of $SIG_I^C(M1)\}_{K_I}$, checks the timestamps and nonce if all the checks are successful then it generates a shared symmetric key K_{ci} between the I and C.
Issuer (I) sends $\{M2, SIG_C^I(M2)\}_{K_c}, Cert_I$ to C containing $M2 = \{PI, phno, K_{ci}, T_i, N_i, N_c\}$ session keys are generated using hashing algorithms with one bit cyclic shift of a master secret each time a session key is generated as shown in [10]. The key set K_{ci} (with {1, 2, 3, .n}) is generated from the secret key K_{ci} and is stored in Mobile Payment Application of the UICC at the client end and in the Issuer.
Step 6: Issuer (I) personalizes Mobile Payment Application on the client's UICC $I \rightarrow C: \{M2, SIG_C^I(M2)\}_{K_c}, Cert_I$ where $2 = \{PI, phno, K_{ci}, T_i, N_i, N_c\}$. Upon receiving the message client checks the authenticity of the message, if the check is successful then it accepts K_{ci} as a shared symmetric key between Issuer and Client.

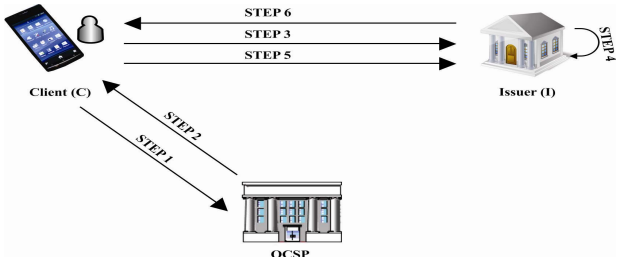

Figure 1: Registration Phase

2) Negotiation Phase

In our proposed framework we employ three mobile agents performing three different tasks: (i) Information gathering and Negotiation Agent (INAg) for brokering and negotiating, which is used in Negotiation phase and (ii) Payment Agent (PAg1 and PAg2) for making payments used in Payment and Deposit phase. INAg is sent to collect the information about goods and the corresponding merchant information and return to client. PAg1 and PAg2 perform payment operations at the Issuer (I) and Payment Gateway (PG). Upon receiving this request, the merchant generates a unique transaction identifier (TID) and sends Order Information (OI), TID, HOI(Hashed OI), MID (Merchant's Identity) and Amt.

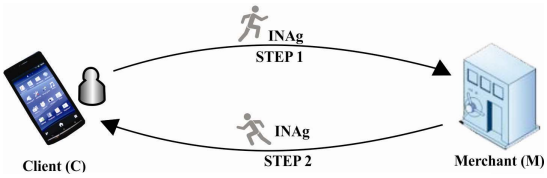

Figure 2: Negotiation Phase

3) Payment Phase

$$Step\ 1: C \rightarrow_{(PAg1)} I: SIG_{C_I}(M3), Pubkey_c$$

$$M3 = PC_{PA1}, (PI)_{K_{ci}}, HOI_C, TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$$

$$M4 = OI, HOI_M, TID_M, MID, N_C, T_C, Amt_M$$

$$C \rightarrow_{(PAg1)} I: M$$

/* $C \rightarrow_{(PAg1)} I: Mx$ means client (C) generates a Payment Agent 1 (PAg1) and sends message with the generated Payment Agent */
Client (C) generates a Payment Agent 1 (PAg1) for sending message Mx to Issuer (I) PAg1carries this information to Issuer (I). Issuer (I) verifies the digital signature, validates the authenticity of the public key and recovers message M3 from $SIG_{C_I}(M3)$.

$$Step\ 2: C \rightarrow_{(PAg2)} PG: SIG_{C_{PG}}(M5), Pubkey_c$$

$$M5 = PC_{PA2}, HPI_C, HOI_C, TID_C, MID, N_C, T_C, Amt_C$$

Client (C) generates a Payment Agent 2 (PAg2) for sending message M5 to Payment Gateway (PG) PAg2 carries this information to Payment Gateway (PG). Payment Gateway (PG) verifies the digital signature, validates the authenticity of the public key and recovers message M5 from $SIG_{C_{PG}}(M5)$.

4. Deposit Phase

$$Step\ 3: I \rightarrow PG: HPI_i, HOI_C, TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$$

The issuer (I) decrypts the message $SIG_{C_I}(M3)$ and gets M3, decrypts PI using the shared symmetric key between the Client and the Bank, checks the PI if found successful it authorizes the PI (Payment Information) and proceeds with the protocol else it aborts the protocol. If the checks are successful it sends $HPI_i, HOI_C, TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$ to Payment Gateway (PG).

$$Step\ 4: PG \rightarrow A: TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$$

The Payment Gateway (PG) receives $SIG_{C_{PG}}(M5)$ from Client (C), decrypts the message $SIG_{C_{PG}}(M5)$ and gets $PC_{PA2}, HPI_C, HOI_C, TID_C, MID, N_C, T_C, Amt_C$. PG also receives $HPI_i, HOI_C, TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$ from issuer (I) through Private Banking Network which is very secure. Payment Gateway (PG) will do the following verifications as given in using the Algorithm 2 from the data it received from Issuer (I) and Client (C) Checks $HPI_i = HPI_c, HOI_C = HOI_i, N_C, T_C$
If all the verifications are found successful then it keeps a copy of the received messages from Issuer (I) and Payment

Gateway (PG) and forwards $TID_C, MID, N_C, T_C, Amt_C, SIG_{C_M}(M4)$ to the Acquirer (A) through Private Banking Network which is very secure.

Algorithm 1: Authorization of Transaction by PG

Algorithm AuthPG {

IF $HOI_C = HOI_M = HOI_I$=TRUE {

So HOI_C sent by C, HOI_I sent by I and HOI_M sent by M are same then PG Authorizes Order Information (OI)}

ELSE {

PG will not Authorize Order Information (OI) }

IF $TID_C = TID_M = TID_I$=TRUE {

So TID_C sent by C, TID_I sent by I and TID_M sent by M are same then PG Authorizes TID }

ELSE {

PG will not Authorize Transaction Identity (TID) }

IF $HPI_C = HPI_i = TRUE$ {

PG Authorizes Payment Information (PI) }

ELSE {

PG will not Authorize Payment Information (PI) }{

PG authorizes the transaction}

ELSE {PG will not authorize the transaction it aborts the Transaction} }

$Step\ 5: A \rightarrow M: SIG_{A_M}(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4))$

A receives $TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4)$ from Payment Gateway (PG) and forwards it to M in the form of $SIG_{A_M}(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4))$.

$Step\ 6: M \rightarrow A: SIG_{M_A}(TID_C, HOI_C, MID_C, Amt_C, AuthOI, HOI_m)$

M receives $SIG_{A_M}(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4))$ from A and decrypts it using his private key and gets $\left(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4)\right)$. Merchant (M) will do the following verifications from the data it received from Acquirer (A).

Checks $HOI_C = HOI_M,\ Amt_c = Amt_m$

If the verifications of the message are successful M authorizes Order Information and sends $SIG_{A_M}(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4))$ to the Acquirer.

$Step\ 7: A \rightarrow PG: TID_C, HOI_C, MID_C, Amt_C, AuthOI, HOI_m)$

Acquirer (A) receives $SIG_{A_M}(TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4))$from M and decrypts it using his private key and gets $TID_C, HOI_C, MID_{,C}, Amt_C, SIG_{C_M}(M4)$. Acquirer (A) forwards $TID_C, HOI_C, MID_C, Amt_C, AuthOI, HOI_m$ message to PG.

$Step\ 8: PG \rightarrow I: TID_C, MID_C, Amt_C, AuthOI$

PG receives $TID_C, HOI_C, MID_C, Amt_C, AuthOI, HOI_m$ from Acquirer (A) and performs all the verifications as given in Algorithm 2 if the verifications are successful it authorizes the transaction and forwards $TID_C, MID_C, Amt_C, AuthOI$ message to Issuer (I).

$Step\ 9: I \rightarrow C: TID, Success/Failure, Amt$

Issuer (I) receives $TID_C, MID_C, Amt_C, AuthOI$ from PG and forwards $TID, Success/Failure, Amt$ to Client (C).

$Step\ 10: A \rightarrow M: TID, Success/Failure, Amt$

Acquirer (A) sends $TID, Success/Failure, Amt$ to Merchant (M).

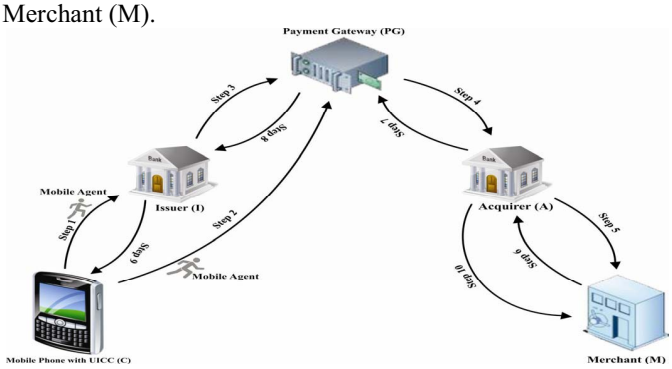

Figure 3: Payment and Deposit Phase

IV. SECURITY ANALYSIS

A. Confidentiality, Authentication, Integrity, Non repudiation and Forward Secrecy

Our proposed EMSET protocol ensures confidentiality, authentication, integrity, non repudiation and Forward Secrecy properties using DSMR mechanism.

B. Order Secrecy & Payment Secrecy

Our proposed EMSET protocol ensures Payment Secrecy and Payment Secrecy. Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)), Issuer will not know about OI thereby achieving Order Secrecy.

C. Key pairs are generated and stored in Tamper resistant device

UICC is the Secure Element used in this framework which is a generic platform for smart card applications. It has been standardized by ETSI EP SCP (ETSI Project Smart Card Platform). The UICC can host a number of different applications, each defining and controlling its own application(s). UICC is personalized by the client and client's credentials are stored in the WIM of UICC. So key pairs are generated and stored in Tamper resistant device (i.e. UICC).

D. For every transaction client needs to agree for a shared symmetric key with PG

In our proposed protocol client need not agree for a shared symmetric key with PG.

E. Execution of Mobile Agent at merchant's TPE

In our proposed protocol mobile agent is not executed at merchant's TPE, it is executed at issuer's TPE.

F. Identity protection from merchant and eavesdropper

In order to prevent a merchant from knowing the identity of Client, an anonymous identity is enrolled by the client with CA and Issuer. CA and Issuer know the real identity of the client. Therefore, as merchant and eavesdropper cannot map the anonymous identity with C'S true identity, client's privacy is protected and untraceable.

G. Transaction Privacy protection from PG and eavesdropper

In our proposed Mobile Payment Protocol (EMSET) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)). PG will not know about OI and PI thereby achieving Transaction privacy from PG. Eavesdropper cannot get OI and PI because the messages are sent using DSMR mechanism thereby achieving Transaction privacy from Eavesdropper.

H. Certificate Validation is needed

Certificate Validation is not needed in our proposed protocol because DSMR is used in our protocol.

I. Proxy based solution

Our proposed protocol is not a proxy based solution.

J. Prevents Double spending, Over spending and Money laundering

Issuer (I) keeps $(PI)K_{ci}, TID, nc, nm, Tm, Tc$ in its archives. If the client or merchant tries to double spend the PI, I can detect this from TID, nc, nm, Tm, Tc so double spending is avoided in EMSET by Issuer. If the client or merchant tries to overspend, I avoids them in doing so since it checks Client (C) funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Issuer is always involved in every transaction thereby preventing money laundering.

K. With stands attacks

Our proposed Mobile Payment protocol (EMSET) withstands the following attacks.

i) Replay Attacks: In EMSET Timestamps and nonce included in the messages exchanged ensures the freshness of the message thereby avoiding replay attacks.

ii) Impersonating attack: An intruder (In) tries to impersonate a client C to CA, which results in CA being cheated. Since In does not have C's private key he fails in doing so. As a result, impersonating attacks fail in our protocol.

iii) Man In The Middle Attack: This attack targets the integrity of the protocol. Our proposed protocol EMSET withstands this attack because the intruder (In) does not have receiver's private key.

V. FORMAL VERIFICATION OF EMSET USING SCYTHER TOOL

We have evaluated our proposed EMSET using the Scyther model checking security protocol verification tool [11]. Scyther is an automatic push-button tool for the verification and falsification of security protocols. EMSET is written using the SPDL (Security Protocol Description Language) and then validated using "Automatic claim" and "Verification claim" procedures in the Scyther tool.

Claim				Status		Comments
EMSETPD	C	EMSETPD,C1	Secret kci	Ok		No attacks within bounds.
		EMSETPD,C2	Secret PI	Ok		No attacks within bounds.
		EMSETPD,C3	Secret OI	Ok		No attacks within bounds.
		EMSETPD,C4	Secret nm	Ok		No attacks within bounds.
		EMSETPD,C5	Secret nc	Ok		No attacks within bounds.
		EMSETPD,C6	Niagree	Ok		No attacks within bounds.
		EMSETPD,C7	Nisynch	Ok		No attacks within bounds.
	M	EMSETPD,M1	Secret nm	Ok	Verified	No attacks.
		EMSETPD,M2	Secret MQ	Ok	Verified	No attacks.
		EMSETPD,M3	Niagree	Ok	Verified	No attacks.
		EMSETPD,M4	Nisynch	Ok	Verified	No attacks.
	A	EMSETPD,A1	Niagree	Ok	Verified	No attacks.
		EMSETPD,A2	Nisynch	Ok	Verified	No attacks.
	PG	EMSETPD,PG1	Niagree	Ok	Verified	No attacks.
		EMSETPD,PG2	Nisynch	Ok	Verified	No attacks.
	I	EMSETPD,I1	Secret kci	Ok	Verified	No attacks.
		EMSETPD,I2	Secret PI	Ok	Verified	No attacks.
		EMSETPD,I3	Niagree	Ok	Verified	No attacks.

Done.

Figure 4: Result of Payment and Deposit phase in EMSET using "Verification Claim" Procedure in Scyther Tool

Claim				Status		Comments
EMSETPD	C	EMSETPD,C8	Secret kci	Ok		No attacks within bounds.
		EMSETPD,C9	Secret nc	Ok		No attacks within bounds.
		EMSETPD,C10	Secret na	Ok		No attacks within bounds.
		EMSETPD,C11	Secret ni	Ok		No attacks within bounds.
		EMSETPD,C12	Secret nm	Ok		No attacks within bounds.
		EMSETPD,C13	Niagree	Ok		No attacks within bounds.
		EMSETPD,C14	Nisynch	Ok		No attacks within bounds.
	M	EMSETPD,M5	Secret kci	Ok	Verified	No attacks.
		EMSETPD,M6	Secret nm	Ok	Verified	No attacks.
		EMSETPD,M7	Secret ni	Ok		No attacks within bounds.
		EMSETPD,M8	Secret nc	Ok	Verified	No attacks.
		EMSETPD,M9	Niagree	Ok	Verified	No attacks.
		EMSETPD,M10	Nisynch	Ok	Verified	No attacks.
	A	EMSETPD,A3	Secret kci	Ok	Verified	No attacks.
		EMSETPD,A4	Secret na	Ok	Verified	No attacks.
		EMSETPD,A5	Secret nm	Ok	Verified	No attacks.
		EMSETPD,A6	Secret nc	Ok	Verified	No attacks.
		EMSETPD,A7	Niagree	Ok	Verified	No attacks.
		EMSETPD,A8	Nisynch	Ok	Verified	No attacks.

Done.

Figure 5: Result of Payment and Deposit phase in EMSET using "Automatic Claim" Procedure in Scyther Tool

VI. COMPARITATIVE ANALYSIS

PROTOCOLS / FEATURES	[1]	[2]	[3]	[4]	[5]	[12]	EM SET
Authentication	YES	YES	YES	YES	YES	YES	YES
Confidentiality	YES	YES	YES	YES	YES	YES	YES
Integrity	YES	YES	YES	YES	YES	YES	YES
Non- Repudiation	YES	NO	NO	YES	YES	NO	YES
Forward Secrecy	NR	NR	NR	NR	NR	NR	YES
Order Secrecy	NR	NR	NR	NR	YES	NR	YES
Payment Secrecy	NR	NR	NR	NR	YES	NR	YES
Key pairs are generated and stored in Tamper resistant device	NO	NO	NO	NO	NO	NO	YES
Are the Signatures generated in "Secure Signature Creation Device (SSCD)"	NO	NO	NO	NO	NO	NO	YES
For every transaction client needs to agree for a shared symmetric key with PG	YES	YES	YES	YES	YES	YES	NO
Execution of Mobile Agent at merchant's TPE	YES	YES	YES	NO	YES	YES	NO
Identity Protection from Merchant	NO	NO	NO	NO	NO	NO	YES
Identity Protection from Eavesdropper	YES	NO	NO	NO	NO	NO	YES
Transaction Privacy Protection from PG & Eavesdropper	YES	NO	NO	NO	NO	NO	YES
Certificate Validation is needed	YES	YES	YES	YES	YES	YES	NO
Proxy based solution	NO	NO	NO	YES	YES	NO	NO
Prevents Double Spending, Over Spending & Money Laundering	NR	NR	NR	NR	NR	NR	YES
Withstands Replay, Impersonation & MITM Attacks	NO	NO	NO	NO	NO	NO	YES
Formal Analysis using SCYTHER TOOL	NO	NO	NO	NO	NO	NO	YES

REFERENCES

[1] Mastercard and Visa. SET Protocol Specifications. http://www.setco.org/set_specifications.html

[2] Romao A. and da Silva M. M., 1998. An Agent-based Secure Internet Payment Systems. Proceedings of TREC'98, LNCS 1402, pp. 80-93.

[3] Wang X. F. et al, 1999, "Secure Agent-Mediated Mobile Payment" Proceedings of PRIMA98, LNAI 1599, pp.162-173.

[4] Supakorn Kungpisdan , Bala Srinivasan , Phu Dung Le, "A Practical Framework for Mobile SET Payment" In Proceedings of the IADIS International E-Society Conference, Lisbon, Portugal, June 3-6 (2003) , pp 321-328.

[5] Chung-Ming Ou, C.R.Ou, "SETNR/A: an agent-based secure payment protocol for mobile commerce", International Journal of Intelligent Information and Database Systems, Vol. 4, No.3, 2010.

[6]OMA, Wireless Application Protocol – Wireless Public Key Infrastructure, WAP-217-WPKI, April 2001.

[7] OMA, WAP Certificate and CRL, WAP-211-X.509, March 2000.

[8] Shiang-Feng Tzenga, Min-Shiang Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem", Computer Standards & Interfaces 26 (2004) 61–71.

[9] Zuhua Shao, "Improvement of digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem", Computer Standards & Interfaces 27 (2004) 61–69.

[10] Supakorn Kungpisdan, Bala Srinivasan, and Phu Dung Le, "Lightweight Mobile Credit-Card Payment Protocol", INDOCRYPT 2003, LNCS 2904, pp. 295–308.

[11] C. J. F. Cremers, "Scyther-Semantics and Verification of Security Protocols," Ph.D. Thesis, Eindhoven University of Technology, 2006.

[12] Xiaolin Pang, Kian–Lee Tan, Yan Wang, and Jian Ren, "A Secure Agent-Mediated Payment Protocol", In: Fourth International Conference on Information and Communications Security (ICICS2002), volume LNCS 2512, Springer-Verlag, pages 422-433.