Electron Commer Res
DOI 10.1007/s10660-015-9192-5

A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography

Shehzad Ashraf Chaudhry[1] · Mohammad Sabzinejad Farash[2] · Husnain Naqvi[1] · Muhammad Sher[1]

Abstract The use of e-payment system for electronic trade is on its way to make daily life more easy and convenient. Contrarily, there are a number of security issues to be addressed, user anonymity and fair exchange have become important concerns along with authentication, confidentiality, integrity and non-repudiation. In a number of existing e-payment schemes, the customer pays for the product before acquiring it. Furthermore, many such schemes require very high computation and communication costs. To address such issues recently Yang et al. proposed an authenticated encryption scheme and an e-payment scheme based on their authenticated encryption. They excluded the need of digital signatures for authentication. Further they claimed their schemes to resist replay, man-in-middle, impersonation and identity theft attack while providing confidentiality, authenticity, integrity and privacy protection. However our analysis exposed that Yang et al.'s both authenticated encryption scheme and e-payment system are vulnerable to impersonation attack. An adversary just having knowledge of public parameters can easily masquerade as a legal user. Furthermore, we proposed improved authenticated encryption and e-payment schemes to overcome weaknesses of Yang et al.'s schemes. We prove the security of our schemes using automated tool ProVerif. The

✉ Mohammad Sabzinejad Farash
sabzinejad@khu.ac.ir

Shehzad Ashraf Chaudhry
shahzad@iiu.edu.pk

Husnain Naqvi
husnain.naqvi@iiu.edu.pk

Muhammad Sher
m.sher@iiu.edu.pk

[1] Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan

[2] Faculty of Mathematics and Computer Sciences, Kharazmi University, Tehran, Iran

Published online: 25 June 2015

improved schemes are more robust and more lightweight than Yang et al.'s schemes which is evident from security and performance analysis.

Keywords Authenticated encryption · E-payment system · Elliptic curve cryptography · Digital signature · Signcryption · ProVerif

1 Introduction

With the rapid development of information and communication technologies, e-commerce has emerged as a viable solution to on line shopping. During recent times the purchase of digital contents has been greatly increased, as per the statistics of U.S. Bureau of census, the on line sale augmented from USD 99.50 billion to USD 343.43 billion during a thirteen years time span. Very similarly china's on line market achieved USD 110.04 billion worth of business despite a number of challenges [1, 2]. Such growth in e-commerce is because of its speed, digitization and accessibility [3]. Electronic payment systems are considered as an integral part of any e-commerce system. Electronic payment systems are categorized into three basic types: business to business (B2B), consumer to consumer (C2C) and business to consumer (B2C). B2C e-payment got popularity after universalization of Internet in early 1990s. A number of B2C payment systems require credit cards for on line payments. With the advent of e-payment systems, the users are having the expediency to save the time and money by using a number of services on line (like payment of bills, purchase of goods etc.).

The primitive e-payment system was proposed by Chaum [4], after then many e-payment systems are proposed [3, 5–10]. While E-commerce is on its way to make daily life more convenient and easy, the main concerns in any e-payment system are security and privacy of participant and contents. The existing e-payment schemes make use of signatures to ensure user's authenticity and message integrity, while they can not ensure user anonymity. Recently Yang et al. [11] pointed out that in signature based schemes sender's signature is generated further the signature is verified on receiver side, this generation and verification of sender's signature burdened the system. Furthermore, the signature is sent on public network which may cause its illegal use. Therefore, Yang et al. [11] proposed a novel authenticated encryption scheme and an e-payment system based on their authenticated encryption scheme. In Yang et al.'s scheme sender make use of his own private key and receiver's public key to form a symmetric key. The same symmetric key is generated by receiver by using his private key. They claimed to achieve the sender authenticity, message confidentiality and user anonymity as the symmetric key can only be generated by legitimate sender and reconstructed by intended legitimate receiver without generating and verifying the sender's signature.

In this paper, we cryptanalyzed Yang et al.'s [11] authenticated encryption scheme and e-payment system and find both to be vulnerable to impersonation attack. We show that an adversary just having the knowledge of public parameters can impersonate as a legitimate user. The attacker can easily exploit the weakness of Yang et al.'s scheme and can fraudulently purchase digital contents by deceiving the bank and merchant. Furthermore, we improved both Yang et al.'s authenticated encryption scheme and e-payment system. We prove the security of our improved schemes using automated tool ProVerif.

Rest of the paper is organized as follows. In Sect. 2, we briefly describe the fundamentals of elliptic curve cryptography along with authenticated encryption and e-payment system. In Sect. 3, we review Yang et al.'s authenticated encryption scheme and its application in e-payment system. In Sect. 4, we performed cryptanalysis of Yang et al.'s authenticated encryption and e-payment schemes. Our improved authenticated encryption scheme is described in Sect. 5.1, while we improve Yang et al.'s e-payment system in Sect. 5.2. We prove the security of our proposed scheme in Sect. 6. In Sect. 7, we performed automated correctness and security verification of our scheme using ProVerif. The performance comparison is shown in Sect. 8. Finally, we conclude in Sect. 9.

2 Preliminaries

In this section, we briefly describe background for elliptic curve cryptography, the basic definitions of authenticated encryption and e-payment systems.

2.1 Elliptic curve cryptography

This sub-section accommodates some of the basic elliptic curve cryptography (ECC) concepts pertinent to this paper. The ECC security has been proved to be more efficient cryptographic scheme as compared to earlier conventional techniques like RSA, DH and DSA [12–21]. This technique provides an equivalent level of security with much less key sizes. The mathematical operations are defined over an elliptic curve equation $E_p(a,b) : y^2 = x^3 + ax + b \bmod p$ where $a, b \in Z_p$ and $4a^3 + 27b^2 \bmod p \neq 0$ and p be a large prime number. Both values a, b defines the elliptic curve, while the points (x, y) that satisfies the former statement including a point at infinity lies on the elliptic curve. The scalar multiplication is defined as repeated addition $tQ = Q + Q + Q..... + Q(t\ times)$, given a point Q and an integer $t \in F_p^*$. All domain parameters like (p, a, b, P, n , h) belong to finite field, F_p^*. E is an abelian group and the point at infinity serves as identity element for this group.

2.2 Authenticated encryption

The concept of authenticated encryption (also termed as signcryption) was first introduced by Zhang et al. [22]. Traditionally authentication [23–25] and confidentiality [26] were considered two distinct tasks and to achieve them the sender first digitally signs the message then performs encryption. Unfortunately this approach is not suitable for resources constrained environments as it double folds the computation and other requirements. Authenticated encryption combines both the processes into a single process to reduce computation, communication and storage costs. An authenticated encryption scheme involves two participants: the sender and the recipient. Initially the sender generates a key, then encrypts the message and generates digital signatures based on message and public key of sender. Finally the sender sends encrypted message and signature tuple to recipient. Upon reception of encrypted message and signature tuple, the recipient generate same key and decrypts the message. Finally the recipient verifies the signatures [27–31].

2.3 E-payment system

An e-payment system facilitates for transecting digital products. A general e-payment system consists of a customer, bank, merchant and a trusted third party to resolve a dispute. The basic aim of an e-payment scheme is to provide framework for on line purchase of digital products while ensuring user's anonymity, fair exchange and dispute resolution. Fair exchange employees that none of the participant should have unfair advantage. In case of any dispute between the participants, the trusted third party is responsible for its resolution. A typical e-payment system is illustrated in Fig. 1. Before making any transaction, all the participants are supposed to register with the system, which in turns assigns a unique identity. Further both merchant and customer must open some account to benefit secure e-payment. The participants are then required to select their private keys and

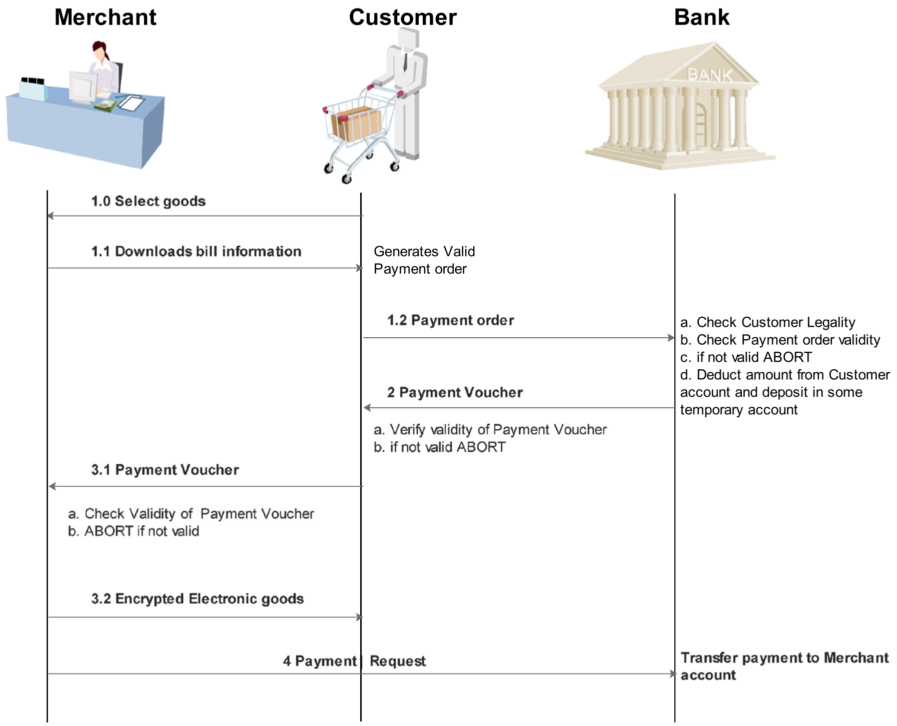
Merchant
Customer
Bank
BANK
1.0 Select goods
1.1 Downloads bill information
Generates Valid Payment order
1.2 Payment order
a. Check Customer Legality
b. Check Payment order validity
c. if not valid ABORT
d. Deduct amount from Customer account and deposit in some temporary account
2 Payment Voucher
a. Verify validity of Payment Voucher
b. if not valid ABORT
3.1 Payment Voucher
a. Check Validity of Payment Voucher
b. ABORT if not valid
3.2 Encrypted Electronic goods
4 Payment Request
Transfer payment to Merchant account

Fig. 1 E-payment system

compute and link their public keys with their bank account. A transactions in e-payment system is consisting of following five phases:

1. *Buying phase* The customer selects his desired goods from merchant's website, then he downloads the bill information from merchant's website. The customer then makes a valid payment order tuple and sends the payment order to the bank,
2. *Paying phase* Upon receiving the payment orders from customer, the bank checks the legality of the customer and validity of the payment order, if legality of the customer is not proved, the session is aborted by the bank. Otherwise the bank deducts bill amount from customer's account and stores the bill amount in some temporary account. Finally bank sends a unique payment voucher with some arbitrary expiry date to the customer.
3. *Exchanging phase* For the received payment voucher, the customer checks its validity. If the voucher is not valid customer aborts the session, otherwise the customer generates a new message tuple based on payment voucher and sends it to the merchant. The merchant after receiving payment voucher checks the customer and voucher legality. The session is aborted if legality is not proved, otherwise merchant sends the encrypted electronic goods to the customer, which upon reception decrypts and use it.
4. *Transferring phase* The merchant sends the payment voucher to bank before expiry date. For the valid payment voucher the bank transfers the voucher amount to merchant's account.
5. *Dispute resolution phase* This is an optional phase and can be committed either by customer or merchant if their arise some dispute among both.

2.4 E-payment security requirements

During e-payment transaction, the financial information is sent over insecure public network so it requires a robust security mechanism which can ensure mutual authentication, confidentiality, integrity, non-repudiation, privacy and prevention of double spending for a single transaction. Following are the requisite security factors to be considered in an e-payment system.

- *Authentication* The customer, bank and merchant should authenticate each other during an e-transaction to avoid false transactions.
- *Confidentiality* The transaction information must be hidden to outsiders, further each of the participant should only know his desired information.
- *Integrity* No one should be allowed to modify the transaction data.
- *Non-repudiation* The participants must not deny their role during a transaction.
- *Privacy protection* Each of the participants should only know his desired information. The bank should know only the amount to be billed not the goods information. Furthermore, the outsider must not know any information regarding the transactions.

- *Double spending prevention* The merchant should be able to use the payment voucher only once. The system must refuse the replay of a previous payment voucher.

3 Review of Yang et al.'s authenticated encryption scheme & e-payment system

This section reviews Yang et al.'s authenticated encryption scheme and its application in e-payment. The scheme is based on elliptic curve cryptography [32–34] further it does not require digital signatures for verification. The scheme and its e-payment version is described in below subsections.

3.1 Yang et al.'s authenticated encryption scheme

Yang et al.'s authenticated encryption scheme consists of three phases initialization, authenticated encryption and verification phases. The notation guide is illustrated in Table 1. Yang et al.'s authenticated encryption scheme is illustrated in Fig. 2 and explained in following subsections:

3.1.1 System initialization phase

During this phase, system selects finite field F_p over a large prime $p \geq 2^{160}$ and an elliptic curve $E_p(a, b)$. Further it selects a base point P in $E_p(a, b)$ and symmetric key algorithm $E_k(.)/D_k(.)$, each legal participant chooses his private key d_i and computes his public key $Y_i = d_i \times P$. Finally system parameters and each participant's public key are published, while each participant keeps his private key secret.

Table 1 Notation guide

Notations	Meaning
p	A large prime number $(p \geq 2^{160})$
$E_p(a, b)$	Selected elliptic curve
P	A base point over $E_p(a, b)$
d_i	Private key of ith legal user
$Y_i = d_i \times P$	Public key of ith legal user
M	Message (plain text)
E_k/D_k	Encryption/decryption
T_i	ith Time stamp
$H(.)$	A one way hash function
$\mathcal{U}_i$	Legal user/customer
$\mathcal{A}$	Adversary
$\mathcal{M}$	Merchant
$\mathcal{B}$	Bank

3.1.2 Authenticated encryption phase

During this phase a legal user $\mathcal{U}_a$ performs authenticated encryption after obtaining another legal user $\mathcal{U}_b$'s public key Y_{ub}. $\mathcal{U}_a$ chooses a random number $r \in Z_q$ and computes $R = r \times Y_{ua}$, $\overline{R} = r \times Y_{ub}$ and $K = d_{ua} \times \overline{R} = (k_x, k_y)$, where d_{ua} is the private key of $\mathcal{U}_a$. Further $\mathcal{U}_a$ computes $C = E_{k_x}(ID_{ua}\|m\|k_x\|T)$. Finally $\mathcal{U}_a$ sends (C, R, T) tuple to $\mathcal{U}_b$.

3.1.3 Verification phase

Upon receiving (C, R, T), $\mathcal{U}_b$ uses his private key d_{ub} to compute $K = d_{ub} \times R = (k_x, k_y)$ then decrypts C using k_x to obtains $(ID_s\|m\|k_x\|T)$. Further it verifies whether T is valid or not. If T is valid then $\mathcal{U}_b$ verifies k_x, if both T and k_x are valid then $\mathcal{U}_b$ consider the message is from legitimate user $\mathcal{U}_a$.

3.2 Yang et al.'s e-payment system

In this subsection, we review Yang et al.'s proposed e-payment system. The e-payment system involves three participants a legal user/customer $\mathcal{U}$, the merchant $\mathcal{M}$ and the bank $\mathcal{B}$. Yang et al.'s scheme is illustrated in Fig. 3 which consists of following five phases:

3.2.1 The initialization phase

In this phase, the system's public parameters are initialized. This phase is analogous to Sect. 3.1.1, where $E_p(a, b)$, $E_k(.)$, $D_k(.)$ and base point P are defined and published. Further $\mathcal{U}$ selects his private key d_u and computes his public key $Y_u = d_u \times P$, similarly $\mathcal{M}$ and $\mathcal{B}$ choose their private keys d_m and d_b, and compute their public keys $Y_m = d_m \times P$ and $Y_b = d_b \times P$. Finally all the participants publish their public keys and keep their private keys secret.

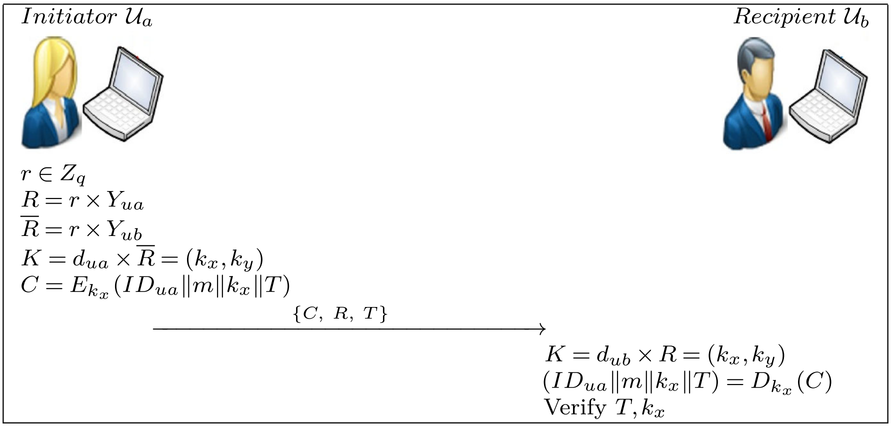
Initiator $\mathcal{U}_a$
Recipient $\mathcal{U}_b$
$r \in Z_q$
$R = r \times Y_{ua}$
$\overline{R} = r \times Y_{ub}$
$K = d_{ua} \times \overline{R} = (k_x, k_y)$
$C = E_{k_x}(ID_{ua}\|m\|k_x\|T)$
$\{C,\ R,\ T\}$
$K = d_{ub} \times R = (k_x, k_y)$
$(ID_{ua}\|m\|k_x\|T) = D_{k_x}(C)$
Verify T, k_x

Fig. 2 Yang et al.'s authenticated encryption scheme

3.2.2 Buying phase

$\mathcal{U}$ initiates the buying phase by first selecting some electronic goods. $\mathcal{U}$ downloads the electronic goods information GI from $\mathcal{M}$'s website then $\mathcal{U}$ selects a random number $r \in Z_q$ and computes $R = r \times Y_u$, $\overline{R} = r \times Y_b$ and $K = d_u \times \overline{R} = (k_x, k_y)$, where k_x is x coordinate of K, while k_y is y coordinate of K. Then $\mathcal{U}$ accumulates the goods payment $p = \sum_{i=1}^{l} price_i$ and computes the payment information as $m = H(GI\|p\|ID_b)$. $\mathcal{U}$ computes $C_1 = E_{k_x}(ID_a\|m\|p\|k_x\|T_1)$ by using k_x. Finally $\mathcal{U}$ sends (C_1, R, T_1) to $\mathcal{B}$, where T_1 is current time stamp.

3.2.3 Paying phase

Upon receiving (C_1, R, T_1) from $\mathcal{U}$, $\mathcal{B}$ computes $K = d_b \times R = (k_x, k_y)$. Then $\mathcal{B}$ uses k_x to decrypt C_1. After decryption $\mathcal{B}$ obtains $(ID_u\|m\|p\|k_x\|T_1) = D_{k_x}(C_1)$. $\mathcal{B}$ further verifies whether T_1 and k_x are valid, if any of these is invalid $\mathcal{B}$ aborts the session. Otherwise, $\mathcal{B}$ deducts amount p from U's account and deposit p into a temporary account. $\mathcal{B}$ further generates the expiry date E and computes $M = m\|E$. Then $\mathcal{B}$ generates his digital signature DS by using his private key d_b and M and stores $\{DS, M\}$ in his database. $\mathcal{B}$ uses k_x and current time stamp T_2 to compute $C_2 = E_{k_x}(DS\|E\|k_x\|T_2)$. Finally $\mathcal{B}$ sends (C_2, T_2) to $\mathcal{U}$. Upon receiving (C_2, T_2), $\mathcal{U}$ decrypts C_2 by using k_x and gets $(DS\|E\|k_x\|T_2) = D_{k_x}(C_2)$. Then $\mathcal{U}$ verifies the validity of k_x and T_2 if any of these is invalid the session is terminated by $\mathcal{U}$. Otherwise, $\mathcal{U}$ accepts the digital signature DS.

3.2.4 Exchanging phase

Initially $\mathcal{U}$ selects a random number $r' \in Z_q$ and computes $R' = r' \times Y_u$, $\overline{R}' = r' \times Y_m$ and $K' = d_u \times \overline{R}' = (k_x', k_y')$. Finally $\mathcal{U}$ using DS as payment proof computes $C_3 = E_{k_x'}(ID_b\|DS\|E\|GI\|k_x'\|T_3)$ and sends (C_3, R', T_3) to $\mathcal{M}$. Upon receiving (C_3, R', T_3), $\mathcal{M}$ computes $K' = d_m \times R' = (k_x, k_y)$. Then $\mathcal{M}$ uses k_x' to decrypt C_3 and obtains $(ID_b\|DS\|E\|GI\|k_x'\|T_3) = D_{k_x'}(C_3)$. $\mathcal{M}$ verifies k_x' and T_2 and aborts the session if any of these are invalid. Otherwise, $\mathcal{M}$ computes goods prices $p = \sum_{i=1}^{l} price_i$, $m = H(GI\|p\|ID_b)$ and $M = m\|E$. $\mathcal{M}$ further verifies DS with M, if digital signature DS proves to be valid $\mathcal{M}$ encrypts electronic goods as $C_4 = E_{k_x'}(Electronic\ goods)$ and sends C_4 to $\mathcal{U}$. Finally $\mathcal{U}$ decrypts C_4 to get desired electronic goods.

3.2.5 Transferring phase

$\mathcal{M}$ sends the payment voucher to $\mathcal{B}$ before expiry date, if $\mathcal{U}$ did not receive the goods, he can ask $\mathcal{B}$ to stop the payment. Otherwise, $\mathcal{B}$ transfers the payment to $\mathcal{M}$'s account from temporary account and deletes $\{DS, M\}$ from his database.

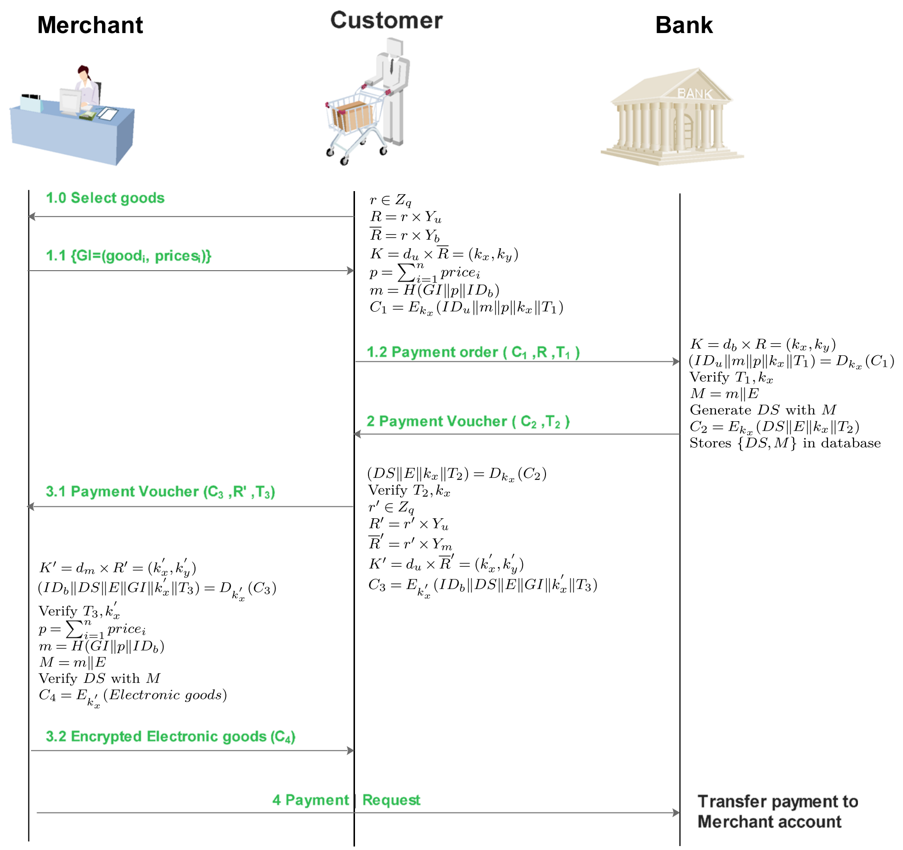
Merchant
Customer
Bank
1.0 Select goods
1.1 {GI=(good$_i$, prices$_i$)}
$r \in Z_q$
$R = r \times Y_u$
$\overline{R} = r \times Y_b$
$K = d_u \times \overline{R} = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI\|p\|ID_b)$
$C_1 = E_{k_x}(ID_u\|m\|p\|k_x\|T_1)$
1.2 Payment order (C$_1$,R ,T$_1$)
$K = d_b \times R = (k_x, k_y)$
$(ID_u\|m\|p\|k_x\|T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m\|E$
Generate DS with M
$C_2 = E_{k_x}(DS\|E\|k_x\|T_2)$
Stores $\{DS, M\}$ in database
2 Payment Voucher (C$_2$,T$_2$)
$(DS\|E\|k_x\|T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$r' \in Z_q$
$R' = r' \times Y_u$
$\overline{R}' = r' \times Y_m$
$K' = d_u \times \overline{R}' = (k_x', k_y')$
$C_3 = E_{k_x'}(ID_b\|DS\|E\|GI\|k_x'\|T_3)$
3.1 Payment Voucher (C$_3$,R' ,T$_3$)
$K' = d_m \times R' = (k_x', k_y')$
$(ID_b\|DS\|E\|GI\|k_x'\|T_3) = D_{k_x'}(C_3)$
Verify T_3, k_x'
$p = \sum_{i=1}^{n} price_i$
$m = H(GI\|p\|ID_b)$
$M = m\|E$
Verify DS with M
$C_4 = E_{k_x'}(Electronic\ goods)$
3.2 Encrypted Electronic goods (C$_4$)
4 Payment Request
Transfer payment to Merchant account

Fig. 3 Yang et al.'s e-payment system

4 Cryptanalysis of Yang et al.'s scheme

This section indicates that authenticated encryption scheme and e-payment system by Yang et al. are vulnerable to impersonation attack. We show that an adversary $\mathcal{A}$ can easily masquerade as a legitimate user by just knowing the public key of the receiver. Before proceeding further, some common assumptions are made as follows:

- $\mathcal{A}$ is having full control over communication channel, $\mathcal{A}$ can intercept, modify, insert or delete any message.
- $\mathcal{A}$ is having access to identities and public keys of communicating parties.

4.1 Impersonation attack on authenticated encryption

We have added Fig. 4 to explain the impersonation attack on authenticated encryption scheme proposed by Yang et al. Let $\mathcal{U}_a$ and $\mathcal{U}_b$ are the two legal users

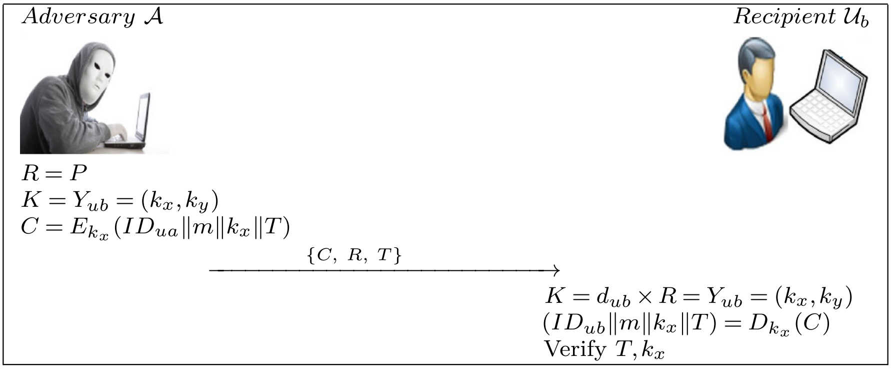
$Adversary\ \mathcal{A}$
$Recipient\ \mathcal{U}_b$
$R = P$
$K = Y_{ub} = (k_x, k_y)$
$C = E_{k_x}(ID_{ua} \| m \| k_x \| T)$
$\{C,\ R,\ T\}$
$K = d_{ub} \times R = Y_{ub} = (k_x, k_y)$
$(ID_{ub} \| m \| k_x \| T) = D_{k_x}(C)$
Verify T, k_x

Fig. 4 Impersonation attack on Yang et al.'s authenticated encryption scheme

and $\mathcal{A}$ be the adversary. $\mathcal{A}$ will perform following steps in-order to masquerade $\mathcal{U}_a$ to deceive the receiver $\mathcal{U}_b$.

Step 1: $\mathcal{A}$ computes following:

$$R = P \tag{1}$$

$$K = Y_{ub} = (k_x, k_y) \tag{2}$$

Step 2: Then $\mathcal{A}$ encrypts the message m along with ID_{ua}, k_x and T, as follows:

$$C = E_{k_x}(ID_{ua} \| m \| k_x \| T) \tag{3}$$

Step 3: $\mathcal{A}$ further sends (C, R, T) tuple to $\mathcal{U}_b$.

Step 4: $\mathcal{U}_b$ upon receiving the tuple (C, R, T), computes $K = (k_x, k_y)$ by using his private key d_{ub} as follows:

$$K = d_{ub} \times R = d_{ub} \times P = Y_{ub} = (k_x, k_y) \tag{4}$$

Step 5: Then $\mathcal{U}_b$ decrypts C by using k_x as follows:

$$(ID_{ua} \| m \| k_x \| T) = D_{k_x}(C) \tag{5}$$

Step 6: $\mathcal{U}_b$ verifies the time stamp T, then checks k_x (decryption key) with k_x received with in decrypted message. If both T and k_x are same $\mathcal{U}_b$ perceive $\mathcal{A}$ as the legitimate $\mathcal{U}_a$.

4.2 Impersonation Attack on e-payment system

Let $\mathcal{U}$ be a legal user, $\mathcal{B}$ be the bank, $\mathcal{M}$ a merchant and $\mathcal{A}$ be the adversary. A will perform steps mentioned in Fig. 5 to masquerade $\mathcal{U}$ to deceive bank $\mathcal{B}$ and merchant $\mathcal{M}$ for fraudulent purchase of electronic goods. These steps are also described below:

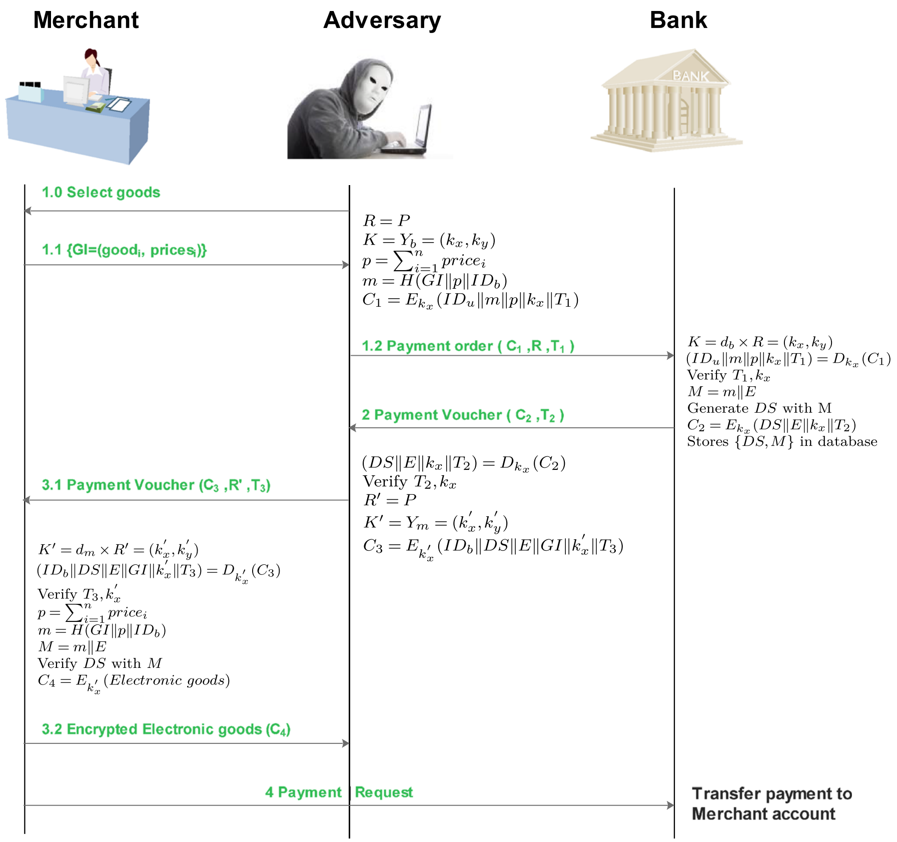
Merchant
Adversary
Bank
1.0 Select goods
1.1 {GI=(good_i, prices_i)}
$R = P$
$K = Y_b = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI\|p\|ID_b)$
$C_1 = E_{k_x}(ID_u\|m\|p\|k_x\|T_1)$
1.2 Payment order (C1 ,R ,T1)
$K = d_b \times R = (k_x, k_y)$
$(ID_u\|m\|p\|k_x\|T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m\|E$
Generate DS with M
$C_2 = E_{k_x}(DS\|E\|k_x\|T_2)$
Stores $\{DS, M\}$ in database
2 Payment Voucher (C2 ,T2)
$(DS\|E\|k_x\|T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$R' = P$
$K' = Y_m = (k_x', k_y')$
$C_3 = E_{k_x'}(ID_b\|DS\|E\|GI\|k_x'\|T_3)$
3.1 Payment Voucher (C3 ,R' ,T3)
$K' = d_m \times R' = (k_x', k_y')$
$(ID_b\|DS\|E\|GI\|k_x'\|T_3) = D_{k_x'}(C_3)$
Verify T_3, k_x'
$p = \sum_{i=1}^{n} price_i$
$m = H(GI\|p\|ID_b)$
$M = m\|E$
Verify DS with M
$C_4 = E_{k_x'}(Electronic\ goods)$
3.2 Encrypted Electronic goods (C4)
4 Payment Request
Transfer payment to Merchant account

Fig. 5 Impersonation attack on Yang et al.'s e-payment system

Step 1: $\mathcal{A}$ selects and downloads the goods information *GI* from $\mathcal{M}$'s website and computes following:

$$R = P \tag{6}$$

$$K = Y_b = (k_x, k_y) \tag{7}$$

$$C_1 = E_{k_x}(ID_u\|m\|k_x\|T_1) \tag{8}$$

Step 2: $\mathcal{A}$ sends $\{C_1,\ R,\ T_1\}$ to $\mathcal{B}$, where T_1 is current time stamp.

Step 3: Upon receiving $\{C_1,\ R,\ T_1\}$, $\mathcal{B}$ computes following:

$$K = d_b \times R = d_b \times P = Y_b = (k_x, k_y) \tag{9}$$

$$(ID_u\|m\|k_x\|T) = D_{k_x}(C_1) \tag{10}$$

Step 4: $\mathcal{B}$ verifies the correctness of T_1 and k_x after performing decryption, if both T_1 and k_x are correct $\mathcal{B}$ generates the expiry date E and $M = m \| E$. Then $\mathcal{B}$ computes digital signature DS with M and:

$$C_2 = E_{k_x}(DS \| E \| k_x \| T_2) \tag{11}$$

Step 5: $\mathcal{B}$ deducts money from $\mathcal{U}$'s account and stores $\{DS, M\}$ in his database. Finally $\mathcal{B}$ sends $\{C_2, T_2\}$ to $\mathcal{U}$, where T_2 is fresh time stamp.

Step 6: $\mathcal{A}$ intercepts the message and use the same key k_x to compute:

$$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2) \tag{12}$$

Step 7: $\mathcal{A}$ verifies T_2 and k_x, then computes:

$$R' = P \tag{13}$$

$$K' = Y_m = (k_x^{'}, k_y^{'}) \tag{14}$$

$$C_3 = E_{k_x'}(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) \tag{15}$$

Step 8: $\mathcal{A}$ sends $\{C_3, R', T_3\}$ to $\mathcal{M}$, where T_3 is fresh time stamp.

Step 9: Upon receiving $\{C_3, R', T_3\}$, $\mathcal{M}$ computes following:

$$K' = d_m \times R' = (k_x^{'}, k_y^{'}) \tag{16}$$

$$(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) = D_{k_x'}(C_3) \tag{17}$$

Step 10: $\mathcal{M}$ verifies the validity of $k_x^{'}$ and T_3, computes following if both are correct.

$$p = \sum_{i=1}^{n} price_i \tag{18}$$

$$m = H(GI \| p \| ID_b) \tag{19}$$

$$M = m \| E \tag{20}$$

Step 11: Further $\mathcal{M}$ computes digital signature DS based on M and checks it's validity by comparing it to the DS obtained in Eq. 17, if it is valid then $\mathcal{M}$ computes:

$$C_4 = E_{k_x'}(Electronic\ goods) \tag{21}$$

Step 12: Finally $\mathcal{M}$ sends encrypted electronic goods C_4 to $\mathcal{U}$.

Step 13: $\mathcal{A}$ intercepts the message and retrieves $Electronic\ goods = D_{k_x'}(C_4)$.

4.3 Discussion on security weakness of Yang et al.'s e-payment scheme

To understand the impact of weakness of Yang et al.'s e-payment scheme, we take an example. Let Bob is an e-payment user with an account in Bank $\mathcal{B}$, he has also initiated his private key and linked his public key with his account. It is well understood that public keys and identities are accessible to any one in the system. Let Alice be an adversary who wants to purchase electronic goods on behalf of Bob. He can impersonate by following the method as described earlier in Subsection 4.2 to deceive bank $\mathcal{B}$ and merchant $\mathcal{M}$.

Alice initially visits $\mathcal{M}$'s website, then selects and downloads the goods and bill information. Alice generates (C_1, R, T_1) tuple as in Eqs. 6, 7 and 8. Alice sends payment order (C_1, R, T_1) to $\mathcal{B}$.

The bank $\mathcal{B}$ upon receiving payment order computes K and C_1, then $\mathcal{B}$ verifies correctness of K_x and T_1 and finds both valid, so $\mathcal{B}$ deducts bill amount from Bob's account and store it in some temporary account. $\mathcal{B}$ computes and sends payment voucher to Alice (apparently Bob). Alice then computes (C_3, R', T_3) as in Eqs. 13, 14 and 15, and sends it to the merchant $\mathcal{M}$.

$\mathcal{M}$ upon reception, computes K' and decrypts C_3 as in Eqs. 16 and 17. Then $\mathcal{M}$ verifies T_3 and k'_x and finds both correct. $\mathcal{M}$ sends electronic goods to Alice (apparently Bob). Finally $\mathcal{M}$ sends received payment voucher to the bank $\mathcal{B}$. The bank transfers the billed amount to $\mathcal{M}$'s account. Hence Alice has purchased electronic goods on behalf of Bob.

5 Proposed authenticated encryption scheme and e-payment system

In following subsections, we describe the proposed authenticated encryption scheme and the improved e-payment system based on proposed authenticated encryption scheme.

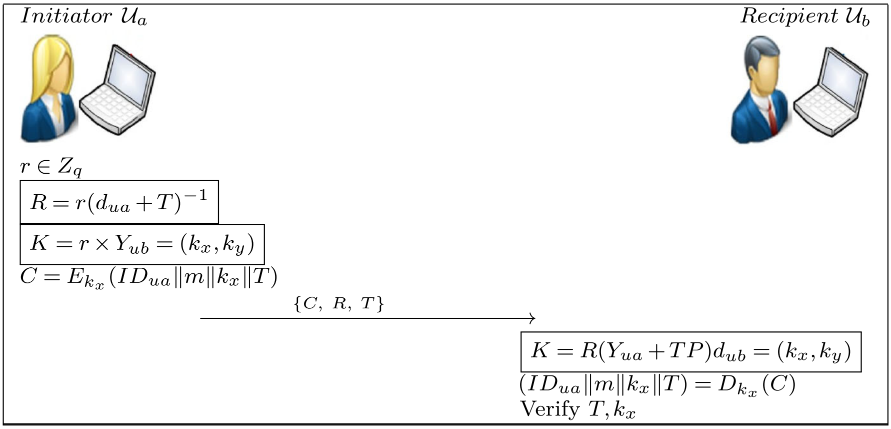
Initiator $\mathcal{U}_a$
Recipient $\mathcal{U}_b$
$r \in Z_q$
$\boxed{R = r(d_{ua} + T)^{-1}}$
$\boxed{K = r \times Y_{ub} = (k_x, k_y)}$
$C = E_{k_x}(ID_{ua} \| m \| k_x \| T)$
$\{C,\ R,\ T\}$
$\boxed{K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y)}$
$(ID_{ua} \| m \| k_x \| T) = D_{k_x}(C)$
Verify T, k_x

Fig. 6 Proposed authenticated encryption scheme

5.1 Proposed authenticated encryption scheme

It can be easily verified that the security weakness of Yang et al.'s scheme was due to the design of R and K, so we just improve the calculations of both of these parameters during authenticated encryption and verification phases while there is no change in initialization phase. Proposed authenticated encryption scheme is shown in Fig. 6 and is also explained in following subsection:

5.1.1 *Authenticated encryption phase*

Authenticated encryption is performed by a legal user $\mathcal{U}_a$ when he wants to send another user $\mathcal{U}_b$ a message m. $\mathcal{U}_a$ performs following steps:

Step 1: $\mathcal{U}_a$ chooses a random $r \in Z_p$ and computes $R = r(d_{ua} + T)^{-1}$ by using his private key d_{ua} and current time stamp T.

Step 2: $\mathcal{U}_a$ further computes $K = r \times Y_{ub} = (k_x, k_y)$, where (k_x, k_y) are x and y coordinates of K respectively.

Step 3: $\mathcal{U}_a$ performs symmetric encryption to compute $C = E_{k_x}(ID_{ua}\|m\|k_x\|T)$ using k_x as common shared key and sends (C, R, T) to $\mathcal{U}_b$.

5.1.2 *Verification phase*

During this phase user $\mathcal{U}_b$ receives (C, R, T), decrypts and verifies that the message is sent by another legitimate user $\mathcal{U}_a$. For verification $\mathcal{U}_b$ performs following steps:

Step 1: $\mathcal{U}_b$ computes $K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y)$ and gets decryption key k_x.

Step 2: $\mathcal{U}_b$ decrypts C using k_x as key to obtain $(ID_{ua}\|M\|k_x\|T)$.

Step 3: $\mathcal{U}_b$ verifies whether the received T and computed k_x are same as they are present in decrypted message, if both are same then surefire it came from real $\mathcal{U}_a$.

5.2 The improved e-payment using proposed scheme

As proved in in Sects. 4.1 and 4.2, Yang et al.'s scheme is vulnerable to impersonation attack, hence not suitable for e-payment system, e-voting and similar applications. We have also improved Yang et al.'s authenticated encryption scheme to work in e-payment systems. The improved e-payment is shown in Fig. 7. The e-payment system is based on proposed authenticated encryption scheme and is consisting of five phases: (1) initialization; (2) buying; (3) paying; (4) exchange; and (5) transferring phases. The detail is as follows:

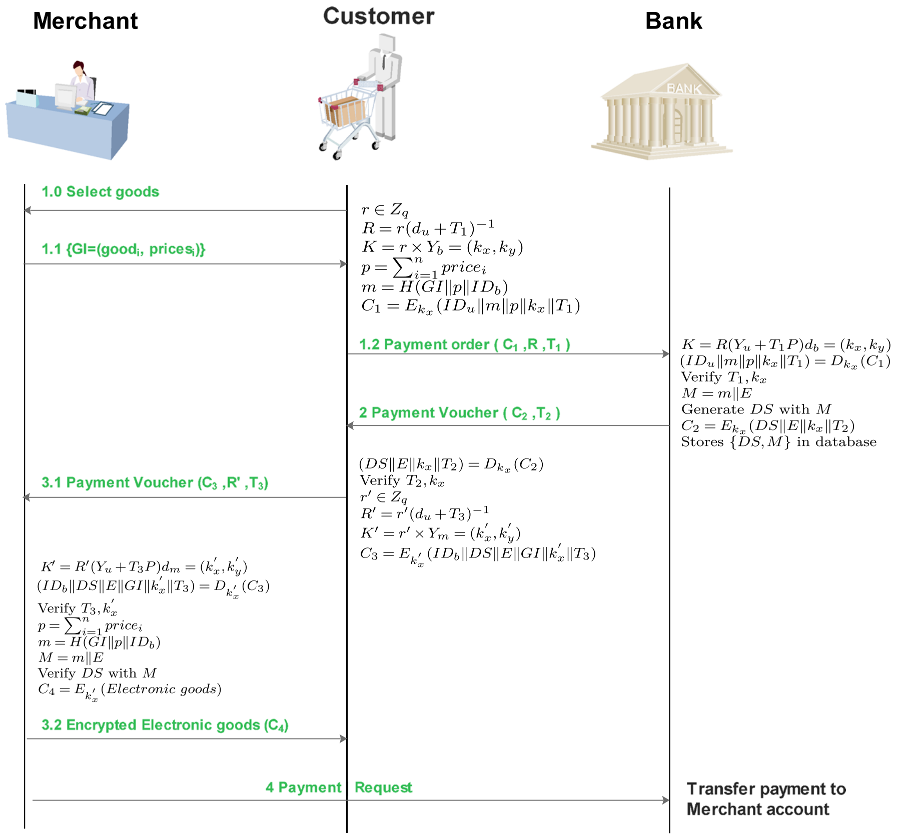
Merchant
Customer
Bank
BANK
1.0 Select goods
1.1 {GI=(good$_i$, prices$_i$)}
$r \in Z_q$
$R = r(d_u + T_1)^{-1}$
$K = r \times Y_b = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI \| p \| ID_b)$
$C_1 = E_{k_x}(ID_u \| m \| p \| k_x \| T_1)$
1.2 Payment order (C$_1$,R ,T$_1$)
$K = R(Y_u + T_1 P)d_b = (k_x, k_y)$
$(ID_u \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m \| E$
Generate DS with M
$C_2 = E_{k_x}(DS \| E \| k_x \| T_2)$
Stores $\{DS, M\}$ in database
2 Payment Voucher (C$_2$,T$_2$)
$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$r' \in Z_q$
$R' = r'(d_u + T_3)^{-1}$
$K' = r' \times Y_m = (k_x^{'}, k_y^{'})$
$C_3 = E_{k_x^{'}}(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3)$
3.1 Payment Voucher (C$_3$,R' ,T$_3$)
$K' = R'(Y_u + T_3 P)d_m = (k_x^{'}, k_y^{'})$
$(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) = D_{k_x^{'}}(C_3)$
Verify $T_3, k_x^{'}$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI \| p \| ID_b)$
$M = m \| E$
Verify DS with M
$C_4 = E_{k_x^{'}}(Electronic\ goods)$
3.2 Encrypted Electronic goods (C$_4$)
4 Payment Request
Transfer payment to Merchant account

Fig. 7 Proposed e-payment system

5.2.1 *Initialization phase*

In this phase the system sets and publishes the public parameters $E_p(a,b), E_k(.), D_k(.), P$, similar to Yang et al.'s scheme as mentioned in Sect. 3.1.1. Each participant 'i' selects his private key d_i and computes and publishes his public key $Y_i = d_i \times P$.

5.2.2 *Buying Phase*

This phase starts when a legal user $\mathcal{U}$ wants to purchase some electronic goods. Initially $\mathcal{U}$ downloads GI (the goods information) from merchant $\mathcal{M}$'s website. Then $\mathcal{U}$ selects a random number $r \in Z_q$, and computes $R = r(d_u + T_1)^{-1}$ and $K = r \times Y_b$. Further $\mathcal{U}$ accumulates the goods price $p = \sum_{i=1}^{n} price_i$ and generates payment text $m = H(GI \| p \| ID_b)$. Finally $\mathcal{U}$ generates $C_1 = E_{k_x}(ID_u \| m \| p \| k_x \| T_1)$ and sends the tuple (C_1, R, T_1) to the bank $\mathcal{B}$.

5.2.3 Paying phase

Upon receiving the authenticated encrypted message tuple (C_1, R, T_1), the bank $\mathcal{B}$, first computes $K = R(Y_u + T_1P)d_b = (k_x, k_y)$ then use k_x to compute $(ID_u\|m\|p\|k_x\|T_1) = D_{k_x}(C_1)$. Further $\mathcal{B}$ checks the validity of time stamp T_1 and verifies whether k_x is same as found after decryption of C_1. $\mathcal{B}$ accepts the message if both T_1 and k_x are valid. Otherwise rejects the message. Further $\mathcal{B}$ deducts the money amounting P from $\mathcal{U}$'s account and transfer p in to a temporary account. $\mathcal{B}$ selects an expiry date E and computes $M = m\|E$. Further $\mathcal{B}$ creates M's digital signature DS based on elliptic curve cryptography as mentioned in [35]. Finally, $\mathcal{B}$ computes and sends $C_2 = E_{k_x}(DS\|E\|k_x\|T_2)$ to $\mathcal{U}$ and stores $\{DS, M\}$ in his database.

5.2.4 Exchange phase

The exchange phase consists of following three steps:

Step 1: $\mathcal{U}$ after receiving encrypted message, first decrypts C_2 to obtain DS and expiry date E. Then $\mathcal{U}$ selects $r' \in Z_q$, and computes $R' = r'(d_s + T_3)^{-1}$, $K = r' \times Y_m = (k_x, k_y)$, $C_3 = E_{k_x}(ID_b\|DS\|E\|GI\|k_x'\|T_3)$. Finally $\mathcal{U}$ sends (C_3, R', T_3) to $\mathcal{M}$.

Step 2: Upon Receiving (C_3, R', T_3), $\mathcal{M}$ computes $K = R'(Y_u + T_3P)d_m = (k_x, k_y)$, then decrypts C_3 using k_x' as decryption key. Then $\mathcal{M}$ verifies validity of T_3 and k_x', if both are valid, $\mathcal{M}$ computes p and $m = H(GI\|p\|ID_b)$. Further $\mathcal{M}$ calculates $M = m\|E$ and verifies the signature DS by using $\mathcal{B}$'s public key, if DS is not valid $\mathcal{M}$ aborts the session. Otherwise, $\mathcal{M}$ computes and sends $C_4 = E_{k_x'}(Electronicgoods)$ to $\mathcal{U}$.

Step 3: $\mathcal{U}$ decrypts C_4 to acquire electronic goods.

5.2.5 Transferring phase

$\mathcal{M}$ sends the payment proof to $\mathcal{B}$ before expiry date. $\mathcal{U}$ is having the facility to ask $\mathcal{B}$ to terminate the transaction if he did not receive the goods, in that case $\mathcal{B}$ transfers back the money from temporary account to $\mathcal{U}$'s account. After expiry date $\mathcal{B}$ transfer the money to $\mathcal{M}$'s account and removes $\{DS, M\}$ from his database.

5.2.6 Dispute resolution phase

If user do not get the desired product or merchant do not get the correct payment voucher then they can initiate dispute resolution phase. A trusted third party is responsible for dispute resolution, in either cases trusted third party will be given the merchant's private key to verify the correctness of key $k_x{}'$. Trusted third party (TTP) after getting message $\{C_3, R', T_3\}$ can verify legality of customer by computing following:

$$K' = R'(Y_u + T_3P)d_m = (k_x^{'}, k_y^{'}) \tag{22}$$

$$(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) = D_{k_x^{'}}(C_3) \tag{23}$$

TTP compares T_3 received in plain text and got after decryption. Similarly TTP compares k_x' computed in Eq. 22 and decrypted in Eq. 23, if both are equal the customer and merchant both are legal. TTP can further verify the encrypted digital signature DS and product's information. Hence TTP can resolve the dispute among both customer and merchant.

6 Security analysis

This section briefly describes the security analysis of our proposed schemes. Table 2 depicts the security comparison of proposed scheme with Yang et al.'s scheme. The improved schemes satisfies all the security requirements mentioned by Yang et al. It is shown that the proposed schemes remain robust even if an adversary intercepts the messages among sender $\mathcal{U}_a$ and receiver $\mathcal{U}_b$. The security of the proposed scheme relies on encryption/decryption key k_x, to generate valid k_x, the adversary $\mathcal{A}$ has to generate valid R. The detailed security analysis is described in following sub sections:

6.1 Replay attack

The adversary $\mathcal{A}$ can replay a past message tuple (C, R, T) as it is to receiver $\mathcal{U}_b$, when $\mathcal{U}_b$ will receive the message, it will first check the validity of time stamp T as T is not fresh, $\mathcal{U}_b$ will realize that message is sent by adversary $\mathcal{A}$ and will simply discard the message.

6.2 Outsider attack

An outsider $\mathcal{A}$ can intercept (C, R, T) of past communication among $\mathcal{U}_a$ and $\mathcal{U}_b$. However he can not succeed in getting m from C as it requires decryption key k_x, which can only be computed as follows:

$$K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y) \tag{24}$$

$\mathcal{A}$ can easily get R, Y_s, T, P but having all these values computing K is ECDLP, as $\mathcal{A}$ is not having private key d_{ub}.

6.3 Impersonation attack

Impersonation attack is only possible if $\mathcal{A}$ can generate valid R and K pair, to generate valid R, $\mathcal{A}$ needs private key of sender $\mathcal{U}_a$. If $\mathcal{A}$ tries to forge R by selecting a random number $\overline{r} \in Z_q$ and computes $\overline{R} = \overline{r}(\overline{d}_{ua} + T)^{-1}$, $\overline{K} = \overline{r} \times Y_{ub}$, then after receiving $(C, \overline{R}, T)$ tuple, $\mathcal{U}_b$ will compute $K = \overline{R}(Y_{ua} + TP)d_{ub}$, which will not be equal to $\overline{r} \times Y_{ub}$. Hence $\mathcal{U}_b$ will be aware that message is sent by $\mathcal{A}$.

6.4 Server spoofing attack

$\mathcal{A}$ can pretend to be a bank server if he can generate $\overline{C_2} = E_{k_x}(\overline{DS} \| \overline{E} \| k_x \| T_2)$ and send $(\overline{C_2}, T_2)$ to $\mathcal{U}_a$. However $\mathcal{A}$ has to obtain d_b to compute $K = R(Y_{ua} + TP)d_b = (k_x, k_y)$ in order to get correct k_x which is infeasible.

6.5 Man-in-middle attack

If $\mathcal{A}$ intercepts the payment information message (C_i, R, T_i) and then replace the time stamp T_i with fresh time stamp T_{fresh}, $\mathcal{U}_b$ after decrypting C_i will compare T_{fresh} with time stamp T_i got after decryption, if both are not same, $\mathcal{U}_b$ will terminate the session. Henceforth, man in middle attack is not viable on proposed schemes.

6.6 ID theft attack

The proposed schemes make use of private and public keys of sender and receiver to generate and verify authenticated message. So if the identity of any or both parties is revealed to the adversary. It will have no effect on security of the scheme.

6.7 Confidentiality

The confidentiality can be broken if $\mathcal{A}$ can decrypt the cipher text C, in proposed scheme all the messages are encrypted by using a symmetric key k_x, it has already been proved in Sect. 6.2 that k_x can only be computed by first getting d_{ub} from Eq. 24, which is an ECDLP. Hence not feasible.

6.8 Authenticity

Proposed schemes ensure the sender's authenticity as receiver extracts k_x by computing $K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y)$, which require $\mathcal{U}_a$'s public key and $\mathcal{U}_b$'s private key, further $\mathcal{U}_b$ verifies the validity of k_x after decryption of C and compares computed k_x and decrypted k_x from cipher text C.

6.9 Integrity

Proposed schemes provide message integrity as if any of the parameter (C, R, T) is modified then receiver $\mathcal{U}_b$ will not be able to verify validity of k_x or T and will simply terminate the session.

6.10 Privacy protection

ID of all the participants are sent in cipher text C, no ID is sent in plain text over public network. Similarly user $\mathcal{U}$ sends GI (goods information) to bank after protecting it by one way hash function $m = H(GI\|p\|ID_b)$. Furthermore, the digital signature DS does not reveal payer's information. Hence buying privacy is protected in proposed scheme.

6.11 Non-repudiation

In proposed e-payment scheme none of the participant can deny the transaction, as trusted third party can check the validity of messages between both customer and merchant as described in Sect. 5.2.6.

6.12 Double spending prevention

The bank keeps $\{DS, M\}$ information in database until payment is transferred to merchant's account. Once payment is transferred to merchant's account, the bank $\mathcal{B}$ deletes the corresponding $\{DS, M\}$ entry. Therefore, $\{DS, M\}$ can be used only once. Hence the proposed scheme prevents double spending of same payment voucher.

7 Protocol verification using ProVerif

We formally verify correctness and robustness of proposed scheme using automated tool ProVerif. Formal verification of security protocols was started in mid 80's by using different methods including logic methods, state space exploration and algebraic methods. Applied π calculus is one of the dominant method belongs to a

Table 2 Security analysis

Scheme→ Security properties↓	Yang et al.	Proposed
Resistance to replay attack	Yes	Yes
Resistance to outsider attack	Yes	Yes
Resistance to impersonation attack	No	Yes
Resistance to server spoofing attack	Yes	Yes
Resistance to man-in-middle attack	No	Yes
Resistance to ID theft attack	No	Yes
Confidentiality	Yes	Yes
Authenticity	No	Yes
Integrity	Yes	Yes
Privacy protection	Yes	Yes
Non-repudiation	No	Yes
Double spending prevention	Yes	Yes

class of algebraic methods. ProVerif is an automated tool which make use of applied π calculus to verify cryptographic protocols [36]. ProVerif can verify trace equivalences like reachability, authentication and secrecy to prove a given protocol cannot reach to a bad state [37]. ProVerif can also be used to prove observational equivalences like anonymity & privacy [38, 39]. For Verification purpose, we model the whole protocol steps according to each participant (User, Merchant, Bank) in ProVerif. Then we check the secrecy of the session key and the reachability property. Finally we got the result as follows:

```
RESULT inj event(endMerchant(id)) ==> inj event(beginMerchant(id)) is true.
RESULT inj event(endBank(id_2234)) ==> inj event(beginBank(id_2234)) is true.
RESULT inj event(endUser(id_4043)) ==> inj event(beginUser(id_4043)) is true.
RESULT not attacker(K[]) is true.
RESULT not attacker(K1[]) is true.
```

The results shows that all the three processes started and terminated successfully. While *not attacker* on both K and $K1$ shows that (1) secrecy of K and $K1$ is true against attacks (2) authentication is satisfied among user and bank as well as between user and merchant.

8 Performance analysis

In this section, we evaluate the performance of proposed scheme by comparing it with Yang et al's scheme [11], before proceeding further, we define some notations as follows:

- T_{pm}: Time for Elliptic curve point multiplication
- T_{sy}: Time for Symmetric encryption/ decryption operation
- T_h: Time for One way hash function

Table 3 illustrates the performance comparison of proposed scheme with Yang et al.'s scheme. The computation time of different cryptographic operations mentioned by Farash [19] are as follows: T_{pm} and T_{sy} takes 0.86 ms and 0.001 ms respectively while time for T_h is negligible. In Yang et al.'s e-payment system the total operation performed by $\mathcal{U}$ are $6T_{pm} + 3T_{sy}$, while $\mathcal{B}$ performs $1T_{pm} + 2T_{sy}$ operations and $\mathcal{M}$ performs $1T_{pm} + 2T_{sy}$ operations. The total computation time taken by $\mathcal{U}$ is 5.163 ms, $\mathcal{B}$ and $\mathcal{M}$ takes 0.862 ms, so total time taken by all participants during execution of Yang et al.'s e-payment system is 6.887 ms. $\mathcal{U}$ in proposed scheme performs $2T_{pm} + 3T_{sy}$ operations, number of operations performed

Table 3 Computation cost analysis

Scheme→ Participant↓	Yang et al.	Proposed
User $\mathcal{U}_A$	$6T_{pm} + 3T_{sy} = 5.163$ ms	$2T_{pm} + 3T_{sy} = 1.723$ ms
Bank $\mathcal{B}$	$1T_{pm} + 2T_{sy} = 0.862$ ms	$1T_{pm} + 1T_{sy} = 0.862$ ms
Merchant $\mathcal{M}$	$1T_{pm} + 2T_{sy} = 0.862$ ms	$1T_{pm} + 2T_{sy} = 0.862$ ms
Total	$8T_{pm} + 7T_{sy} = 6.887$ ms	$4T_{pm} + 7T_{sy} = 3.447$ ms

by $\mathcal{B}$ are $1T_{pm}+2T_{sy}$, while $\mathcal{M}$ performs $1T_{pm}+2T_{sy}$ operations. Total time taken by $\mathcal{U}$ in proposed scheme is 1.723 ms, which is roughly one third of the time taken by $\mathcal{U}$ in Yang et al.'s scheme. While $\mathcal{B}$ and $\mathcal{M}$ takes 0.862 ms, which are equal to time taken by both $\mathcal{B}$ and $\mathcal{M}$ in Yang et al.'s scheme, total time taken by all participants during execution of proposed e-payment system is 3.447 *ms* as shown in Table 3. Hence in proposed scheme user $\mathcal{U}$ takes approximately 66% less computation time as compared with Yang et al.'s scheme. Therefore proposed scheme provides more robustness against attacks and is more lightweight as compared with Yang et al.'s scheme.

9 Conclusion

In this paper, we cryptanalyzed Yang et al.'s authenticated encryption and e-payment schemes. We proved that both of Yang et al.'s schemes are vulnerable to impersonation attack. As remedy, we proposed improved authenticated encryption scheme to overcome security weaknesses of Yang et al.'s scheme. Furthermore, we also improved e-payment system of Yang et al. We have performed informal and formal verification of our improved protocol using widespread automated tool ProVerif. The proposed schemes ensured robustness against all known attacks, while reducing about 66% computation cost on user side as compared to Yang et al.'s scheme. Hence proposed scheme improved the security as well as computation overhead and is more suitable for resource constrained environments.

Acknowledgments Authors would like to thank the anonymous reviewers and Prof. John Gerdes for their valuable suggestions to improve the quality, correctness, presentation and readability of the manuscript.

Appendix

In this appendix, we provided the ProVerif verification code for the proposed e-payment system. The protocol model of ProVerif is consisting of three parts. In declaration cryptographic primitives are defined as constructors, destructors and equations. Names are also defined in declaration part. Processes and sub processes are defined in process part, while the protocol is modeled in main process part. In ProVerif, cryptographic primitives are represented as set of functions (termed as constructors and destructors), further ProVerif make use of equations to represent algebraic relations like Diffie-Hellman key agreement. We modeled the proposed scheme as parallel execution of three distinct processes namely user U, Merchant M and the Bank B as defined below:

```
process ((!pUser) | (!pBank) | (!pMerchant))
```

The attacker is modeled by the predicate *attacker(X)*, where X is not known to attacker, if the predicate *not attacker(X)* results into false, then protocol secrecy and authentication is not maintained, otherwise protocol is secure. The attacker knows all public parameters like participants public keys and other related terms. The proposed protocol is modeled as set of steps mentioned in Sect. 5.2, and shown in Fig. 7, in beginning two public channels are defined: *ch1* for communication between the user and bank, while *ch2* for communication between user and merchant.

```
(**********************channels*********************)
free ch1:channel.    (* U to B *)
free ch2:channel.    (* U to M *)
```

The constants and variables are defined as:

```
(***************constants & variables***************)
const p: bitstring.
const P: bitstring.
free GI: bitstring.
free Db: bitstring [private].
free Du: bitstring [private].
free Dm: bitstring [private].
const IDu: bitstring.
const IDb: bitstring.
const IDm: bitstring.
```

where *Du, Db* and *Dm* are private keys of respective participants, while *IDu,IDb* and *IDm* are public identities of user, bank and merchant respectively. *P* is the base point selected over elliptic curve $E_p(a,b)$ and *p* is the price of goods while *GI* is the goods informations. The constructors, destructors and equations are defined as follows:

```
(*******************constructors********************)
fun consset(bitstring,bitstring):bitstring.
fun add(bitstring,bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun syme(bitstring,bitstring):bitstring.(*encryption*)
fun inverse(bitstring):bitstring.
fun getx(bitstring):bitstring.(*get x coordinate*)
fun sig(bitstring):bitstring.(*signature*)
fun h(bitstring):bitstring.(*hash*)
(**************destructors & equations**************)
reduc forall m:bitstring,key:bitstring;
symd(syme(m, key),key)=m.(*decryption*)
equation forall a:bitstring; inverse(inverse(a))=a.
```

Events for user, bank and merchant are defined as follows:

```
(**********************events***********************)
event beginUser(bitstring).
event endUser(bitstring).
event beginBank(bitstring).
event endBank(bitstring).
event beginMerchant(bitstring).
event endMerchant(bitstring).
```

There are three distinct processes in proposed protocol: user, bank and merchant. The user process computes *R*, *K*, *p*, *m* and *C*1 and sends $\{C1, R, T1\}$ to bank using channel *ch1*. After then user receives *C*2 and *T*2 from bank and verifies *T*2 and *kx* finally user computes *R*, *K*1 and *C*3 and sends $\{C3, R1, T3\}$ to merchant on channel *ch2*. The user process is modeled as follows:

```
(**********************User Process***********************)
let pUser=
let Yu=mult(Du,P) in
out(ch1,(Yu));
out(ch2,(Yu));
in(ch1,(XYb:bitstring));
in(ch2,(XYm:bitstring));
new r:bitstring;
new T1: bitstring;
event beginUser(IDu);
let R=mult(r,inverse(add(Du,T1))) in
let K=mult(r,XYb) in
let m=h(consset(GI,consset(p,IDb))) in
let C1=syme((IDu,m,p,getx(K),T1),getx(K)) in
out(ch1,(C1,R,T1)); (*To bank*)
in(ch1,(XC2:bitstring,XT2:bitstring));
let (XDs:bitstring,XE:bitstring,XXkx:bitstring,XXT2:bitstring) =
    symd(XC2,getx(K)) in
if(XT2=XXT2)
then
if(getx(K)=XXkx)
then
new r1: bitstring;
new T3: bitstring;
let R1=mult(r1,inverse(add(Du,T3))) in
let K1=mult(r1,XYm) in
let C3=syme((IDb,XDs,XE,GI,getx(K1),T3),getx(K1)) in
out(ch2,(C3,R1,T3)); (*To merchant*)
event endUser(IDu).
```

The bank process after receiving $\{C1, R, T1\}$, first computes K and decrypts $C1$ using x coordinates of K, then bank verifies validity of Kx and $T1$. Finally bank computes M, E, DS and $C2$ and sends $\{C2, T2\}$ to user via channel *ch1*. The bank process is described as follows:

```
(**********************Bank***********************)
let pBank=
let Yb= mult(Db,P) in
in(ch1,(XYu:bitstring));
out(ch1,(Yb));
in(ch1,(XC1:bitstring,XR:bitstring,XT1:bitstring));
event beginBank(IDb);
let K=mult(mult(XR,add(XYu,mult(XT1,P))),Db) in
let (=IDu, Xm:bitstring, Xp:bitstring, Xkx:bitstring, XXT1:bitstring
    ) = symd(XC1,getx(K)) in
if(getx(K)=Xkx)
then
if(XT1=XXT1)
then
new E: bitstring;
new T2: bitstring;
let M=consset(Xm,E) in
let DS=sig(M) in
let C2=syme((DS,E,Xkx,T2),Xkx) in
out(ch1,(C2,T2));
event endBank(IDb).
```

The merchant process receives $\{C3, R1, T3\}$ from user. Merchant process further computes *K1* and then perform symmetric decryption on *C1* using *K1x*, which is *x* coordinate of *K1*. Further merchant verifies validity of *T3* and *K1x* and computes *p,m,M* and checks the signatures *DS* with *M*. Finally merchant process sends encrypted good *C4* to user. The merchant process is as follows:

```
(********************Merchant**********************)
let pMerchant=
in(ch2,(XYu:bitstring));
let Ym= mult(Dm,P) in
out(ch2,(Ym));
in(ch2,(XC3:bitstring,XR1:bitstring,XT3:bitstring));
event beginMerchant(IDm);
let K1=mult(mult(XR1,add(XYu,mult(XT3,P))),Dm) in
let (=IDb, XDs:bitstring, XE:bitstring,XGI:bitstring,
Xkx:bitstring,XXT3:bitstring)=symd(XC3,getx(K1)) in
if(XT3=XXT3)
then
if(getx(K1)= Xkx)
then
let m=h(consset(GI,consset(p,IDb))) in
let M=consset(m,XE) in
if(sig(M)=XDs)
then
let C4=syme(GI,getx(K1)) in
out(ch2, (C4));
event endMerchant(IDm).
```

The parallel execution of three processes are modeled as:

```
process ( (!pUser) | (!pBank) | (!pMerchant) )
```

The attacker query is applied on two secret keys *K* and *K1* as follows:

```
(*********************query**********************)
free K1:bitstring [private].
free K:bitstring [private].
query attacker(K1).
query attacker(K).
query id:bitstring;inj_event(endUser(id))==>inj_event(beginUser(id))
query id:bitstring;inj_event(endBank(id))==>inj_event(beginBank(id))
query id:bitstring;inj_event(endMerchant(id))==>inj_event(
    beginMerchant(id)).
```

References

1. Chen, S., & Ning, J. (2002). Constraints on e-commerce in less developed countries: The case of china. *Electronic Commerce Research*, *2*(1–2), 31–42. doi:10.1023/A:1013331817147.
2. Kshetri, N. (2013). Cybercrime and cyber-security issues associated with china: some economic and institutional considerations. *Electronic Commerce Research*, *13*(1), 41–69. doi:10.1007/s10660-013-9105-4.
3. Huang, X., Dai, X., & Liang, W. (2014). Bulapay: A novel web service based third-party payment system for e-commerce. *Electronic Commerce Research*, *14*(4), 611–633. doi:10.1007/s10660-014-9172-1.
4. Chaum, D. (2013). Blind signatures for untraceable payments. In *Advances in cryptology—CRYPTO '86 Proceedings* (pp. 199–203). Berlin: Springer.

5. Lysyanskaya, A., & Ramzan, Z. (1998). Group blind digital signatures: A scalable solution to electronic cash. In D. M. Goldschlag & S. G. Stubblebine (Eds.), *Financial cryptography* (pp. 184–197). Berlin: Springer.
6. Zhang, L., Zhang, F., Qin, B., & Liu, S. (2011). Provably-secure electronic cash based on certificateless partially-blind signatures. *Electronic Commerce Research and Applications*, *10*(5), 545–552.
7. Xiaojun, W. (2010). An e-payment system based on quantum group signature. *Physica Scripta*, *82*(6), 65403.
8. Eslami, Z., & Talebi, M. (2011). A new untraceable off-line electronic cash system. *Electronic Commerce Research and Applications*, *10*(1), 59–66.
9. Yen, Y.-C., Wu, T.-C., Lo, N.-W., & Tsai, K.-Y. (2012). A fair-exchange e-payment protocol for digital products with customer unlinkability. *KSII Transactions on Internet and Information Systems*, *6*(11), 2956–2979.
10. Chen, X., Li, J., Ma, J., Lou, W., & Wong, D. S. (2014). New and efficient conditional e-payment systems with transferability. *Future Generation Computer Systems*, *37*, 252–258.
11. Yang, J.-H., Chang, Y.-F., & Chen, Y.-H. (2013). An efficient authenticated encryption scheme based on ecc and its application for electronic payment. *Information Technology And Control*, *42*(4), 315–324.
12. Farash, M. S., & Attari, M. A. (2014). A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. *The Journal of Supercomputing*, *69*(1), 395–411.
13. Irshad, A., Sher, M., Faisal, M. S., Ghani, A., Ul Hassan, M., & Ch, S. A. (2014). A secure authentication scheme for session initiation protocol by using ecc on the basis of the tang and liu scheme. *Security and Communication Networks*, *7*(8), 1210–1218.
14. Irshad, A., Sher, M., Rehman, E., Ch, S. A., Ul Hassan, M., & Ghani, A. (2013). A single round-trip sip authentication scheme for voice over internet protocol using smart card. *Multimedia Tools and Applications*. doi:10.1007/s11042-013-1807-z.
15. Farash, M. S., & Attari, M. A. (2013). An enhanced authenticated key agreement for session initiation protocol. *Information Technology and Control*, *42*(4), 333–342.
16. Farash, M. S. (2014). Cryptanalysis and improvement of an efficient mutual authentication rfid scheme based on elliptic curve cryptography. *The Journal of Supercomputing*, *70*(1), 987–1001.
17. Farash, M. S., & Attari, M. A. (2014). An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. *International Journal of Communication Systems*. doi:10.1002/dac.2848.
18. Farash, M. S. (2014). Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. *Peer-to-Peer Networking and Applications*. doi:10.1007/s12083-014-0315-x.
19. Farash, M. S. (2015). Cryptanalysis and improvement of an improved authentication with key agreement scheme on elliptic curve cryptosystem for global mobility networks. *International Journal of Network Management*, *25*(1), 31–51.
20. Farash, M. S., Kumari, S., & Bakhtiari, M. (2015). Cryptanalysis and improvement of a robust smart card secured authentication scheme on sip using elliptic curve cryptography. *Multimedia Tools and Applications*. doi:10.1007/s11042-015-2487-7.
21. Farash, M. S., Islam, S. H., & Mohammad, S. O. (2015). A provably secure and efficient two-party password-based explicit uthenticated key exchange protocol resistance to password guessing attacks. *Concurrency and Computation: Practice and Experience*. doi:10.1002/cpe.3477.
22. Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption)$\langle\langle$ cost (signature) + cost (encryption). In *Advances in Cryptology-CRYPTO'97* (pp. 165–179). Berlin: Springer.
23. He, D., Kumar, N., & Chilamkurti, N. (2015). A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks, Information Sciences. doi:10.1016/j.ins.2015.02.010
24. He, D., & Zeadally, S. (2015). Authentication protocol for an ambient assisted living system. *Communications Magazine, IEEE*, *53*(1), 71–77.
25. Chaudhry, S., Naqvi, H., Shon, T., Sher, M., & Farash, M. (2015). Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. *Journal of Medical Systems*, *39*(6), 1–11. doi:10.1007/s10916-015-0244-0.
26. Abdalla, M., Benhamouda, F., & Pointcheval, D. (2015). Public-key encryption indistinguishable under plaintext-checkable attacks. In *Public-Key Cryptography—PKC 2015* (pp. 332–352). Berlin: Springer.

27. Ch, S. A., Nizamuddin, N., Sher, M., Ghani, A., Naqvi, H., & Irshad, A. (2014). An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography. *Multimedia Tools and Applications*. doi:10.1007/s11042-014-2283-9.
28. Ch, S. A., Nizamuddin, N., & Sher, M. (2012). Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In *Information systems, technology and management* (pp. 135–142). Springer.
29. Nizamuddin, N., Ch, S. A., Nasar, W., & Javaid, Q. (2011.)Efficient signcryption schemes based on hyperelliptic curve cryptosystem. In *2011 7th IEEE international conference on emerging technologies (ICET)* (pp. 1–4).
30. Nizamuddin, N., Ch, S. A., & Amin, N. (2011). Signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In *IEEE high capacity optical networks and enabling technologies (HONET), 2011* (pp. 244–247).
31. Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption) cost (signature) + cost (encryption). In *Advances in cryptology-CRYPTO'97* (pp. 165–179). Santa Barbara: Springer.
32. Li, C.-T. (2011). Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control*, *40*(2), 157–162.
33. Hong, J.-W., Yoon, S.-Y., Park, D.-I., Choi, M.-J., Yoon, E.-J., & Yoo, K.-Y. (2011). A new efficient key agreement scheme for vsat satellite communications based on elliptic curve cryptosystem. *Information Technology and Control*, *40*(3), 252–259.
34. Farash, M. S., & Attari, M. A. (2014). A provably secure and efficient authentication scheme for access control in mobile pay-tv systems. *Multimedia Tools and Applications*. doi:10.1007/s11042-014-2296-4.
35. Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, *1*(1), 36–63.
36. Xie, Q., Dong, N., Tan, X., Wong, D. S., & Wang, G. (2013). Improvement of a three-party password-based key exchange protocol with formal verification. *Information Technology And Control*, *42*(3), 231–237.
37. Xie, Q., Dong, N., Wong, D. S., & Hu, B. Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol. *International Journal of Communication Systems*. doi:10.1002/dac.2858
38. Hu, B., Xie, Q., & Li, Y. (2011). Automatic verification of password-based authentication protocols using smart card. In *2011 IEEE international conference on information technology, computer engineering and management sciences (ICM)* (Vol. 1, pp. 34–39).
39. Cheval, V., & Blanchet, B. (2013). Proving more observational equivalences with proverif. In D. Basin & J. C. Mitchell (Eds.), *Principles of security and trust* (pp. 226–246). Berlin: Springer.

Shehzad Ashraf Chaudhry received his MS Computer Science with distinction, from International Islamic University Islamabad, Pakistan in 2009 and was awarded *Gold Medal*. Currently he is working as Lecturer at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. He authored more than 20 scientific publications published in different international journals and proceedings. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, E- Payment systems, MANETs, SIP authentication, IP Multimedia subsystem and Next Generation Networks.

Mohammad Sabzinejad Farash received the B.Sc. degree in Electronic Engineering from Shahid Chamran College of Kerman in 2006, and the M.Sc. degree in Communication Engineering from I. Hussein University in 2009. He also received the Ph.D. degree in Cryptographic Mathematics at the Department of Mathematics and Computer Sciences of Tarbiat Moallem University in Iran in 2013. His research interests are Security Protocols and Provable Security Models.

Husnain Naqvi received his Ph.D. from The University of Auckland, New Zealand. Currently he is working as Assistant Professor at the Department of Computer Science, International Islamic University, Islamabad. He authored more than 30 scientific publications published in different international journals and proceedings. His broad research interests include Sensor Networks, Collaborative Communications, Lightweight Cryptography, Beamforming and Space Time Block Codes.

Muhammad Sher is a Professor in international Islamic University, Islamabad, Pakistan. He is having more than 120 scientific publications. He received his Ph.D. Computer Science from TU Berlin, Germany and M. Sc. from Quaid-e-Azam University, Islamabad. His research interests include Next Generation Networks, IP Multimedia Sub-systems and Network Security.