ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2015, T. 44, Nr. 4

An Improved Authentication Scheme for Electronic Payment Systems in Global Mobility Networks

Mohammad Heydari, S. Mohammad-Sajad Sadough

Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran
e-mail: m_heydari@sbu.ac.ir, s_sadough@sbu.ac.ir

Shehzad Ashraf Chaudhry

Department of Computer Science & Software Engineering, International Islamic University
Islamabad, Pakistan
e-mail: shahzad@iiu.edu.pk

Mohammad Sabzinejad Farash

Department of Mathematics and Computer Sciences, Kharazmi University
Tehran, Iran
e-mail: sabzinejad@khu.ac.ir

Mohammad Reza Aref

Department of Electrical Engineering, Sharif University
Tehran, Iran
e-mail: aref@sharif.edu

crossref http://dx.doi.org/10.5755/j01.itc.44.4.9197

Abstract. Recently Yang et al. proposed an authenticated encryption scheme based on elliptic curve cryptography. The scheme reduced computation cost by excluding the construction of sender's digital signatures. Furthermore, Yang et al. presented an e-payment system based on their authenticated encryption scheme. They claimed their scheme to resist replay, man-in-middle, impersonation and identity theft attack, while providing confidentiality, authenticity, integrity and privacy protection. However, in this paper we show that Yang et al.'s both authenticated encryption scheme and e-payment system are vulnerable to impersonation attack. An attacker after acquiring the public key and identities of the participants can easily masquerade as legitimate user. Then, we presented improvements over both Yang et al.'s authenticated encryption and e-payment schemes. We analyze the security of proposed schemes using widespread automated tool ProVerif. The proposed schemes are more secure and lightweight as compared with Yang et al.'s schemes.

Keywords: authenticated encryption; e-payment system; elliptic curve cryptography; digital signature; signcryption.

1. Introduction

Authentication and message confidentiality are the main requirements in-order to secure resource constrained environment. Till now many authentication [1-14] and encryption [15-17] schemes were proposed. Traditionally, authentication and confidentiality 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. Therefore, a single process, which combines both authentication and encryption is of indispensable need. Zhang [18] was the premier to introduce notion of authenticated encryption (some times termed as signcryption). Till then a lot of authenticated encryption schemes were proposed [19-26].

An authenticated encryption consists of three phases: (1) key generation phase; (2) authenticated encryption phase; and (3) verification phase. Zhang [18] proposed an authenticated encryption scheme based on the notion of ElGamal cryptosystem. The scheme was proved to be very efficient when compared with sign-then-encryption strategy on the same ElGamal cryptosystem. However, Zhang's scheme was inefficient and lacking many security features but it provided a basis for future research. Bao and Deng [27] then proposed another direct verifiable authenticated encryption scheme. Their scheme provided the facility to verify the validity of plain text, which is a serious threat to confidentiality. Gamage et al.[28] then proposed another authenticated encryption scheme to improve Bao and Deng' scheme. Gamage et al.'s scheme provided the facility to verify a cipher message. Their scheme was lacking forward secrecy. Zheng and Imai [29] were the first to define authenticated encryption scheme based on elliptic curve cryptography. Elliptic curve got attention during recent past because of its low resource requirements for the same level of security as compared with traditional public key cryptosystems. However, their scheme was lacking forward secrecy and public verifiability. Hwang et al. [30] then proposed another authenticated encryption scheme. Hwang et al.'s scheme was more efficient as compared with previous schemes but their scheme lacks confidentiality during verification of message. Toorani and Shirazi [31] proposed another scheme which was more efficient than Hwang et al.'s scheme and was more secure than previous schemes. Similarly Yoon et al.[32] also proposed efficient authenticated encryption scheme.

Motivation and Contributions: Recently Yang et al. [33] pointed out that in existing authenticated encryption schemes [19-25, 27-32] sender's signature is generated further the signature is verified on receiver side, this generation and verification of sender's signature burdened the system. They also emphasized that the signature transmitted over insecure public network may result into its illegal use. Yang et al. [33] then proposed a novel authenticated encryption scheme without sender's signature. In their proposed scheme [33], the sender generates a symmetric key based on his own private key and the public key of the receiver. The receiver generates the same symmetric key based on his private key and public key of the sender. Yang et al. [33] further asserted that their proposed scheme achieved authenticity and confidentiality. The security of their scheme is relied on generation and reconstruction of shared symmetric key, which they claimed can only be accomplished by legitimate sender and receiver respectively. As an application, Yang et al. presented an e-payment system using their authenticated encryption scheme. However, in this paper, we show that Yang et al.'s authenticated encryption scheme is insecure against sender impersonation attack. We prove that an attacker after acquiring the public keys and identities of sender and receiver can easily masquerade as a legal sender. We also show that the Yang et al.'s e-payment system [33] is also suffering from the same weakness, where an adversary can masquerade as a legitimate customer and can redeem electronic products on behalf of legal customer while cheating the bank and merchant. Then we propose an improved authenticated encryption scheme and e-payment system.

Contributions of this paper are threefold. Initially we prove that Yang et al.'s authenticated encryption and e-payment schemes are vulnerable to impersonation attacks. Secondly, we propose improved authenticated encryption and e-payment schemes. Both of our schemes are robust against all known attacks. Furthermore, our proposed schemes are more lightweight than Yang et al.'s schemes. Finally, we prove the security of our proposed schemes using the widely accepted automated tool ProVerif.

2. Preliminaries

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

2.1. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is based on some non-singular elliptic curve $E_p(a,b)\colon y^2 = x^3 + ax + b\text{mod}p$ over finite field F_p which is deliberated as the set of points (x,y) over $E_p(a,b)$, where $a, b€ \in Z_p$ & $4a^3 + 27b^2\text{mod}p \neq 0$ and p is a large prime number, along with a point at infinity to serve as identity element for the group. There are two primitive operations, point addition and scalar multiplication. The latter is defined as repeated addition $xP = P + P + P\ldots\ldots + P$ $(x\ times)$, given a point P and an integer $x \in F_p$. ECC provides identical security as compared with existing other cryptographic methods like DH, RSA, DSA with shorter parameters and lowered resources [8, 10, 34, 35].

2.2. Authenticated Encryption

Zhang [18] was the premier to introduce the notion of authenticated encryption. Before that authentication and confidentiality were deliberated to be the distinct tasks. To maintain user authenticity and message integrity and confidentiality, the sender primarily signs the message using his private key and then encrypts the message using receiver's public key. This method, is however, not suitable for resource constrained devices. Such twofold operations perform paired computation and consume related resources like memory and communication power. Fortunately, the authenticated encryption notion put forward by Zhang [18] can avoid such resource hungry tasks. In an authenticated encryption scheme, the sender first generates a symmetric key, then performs encryption and computes signatures based on message and receiver's public key and sends both to receiver. On receiving side, the receiver generates the same symmetric key, then decrypts the message and checks signature validity [21-24].

2.3. E-payment system

An e-payment system facilitates the digital transactions. 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 participants 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 as shown in Fig. 2.2 consists of the following five phases:

1. Buying phase: A customer initiates this phase by downloading the bill information from merchant's website, and then asks the bank for payment voucher.
2. Paying phase: During this phase bank deducts the bill amount from customer's account and stores the bill amount in some temporary account, finally bank generates and sends payment voucher with some arbitrary expiry date to customer.
3. Exchanging phase: During this phase the customer and merchant exchange the payment voucher and digital product to each other.
4. Transferring phase: The merchant sends the payment voucher to bank before expiry date, the bank transfers the voucher amount to merchant's account.
5. Dispute resolution phase: This is an optional phase, and can be initiated either by customer or merchant if their arise some dispute among both.

2.4. Adversary Model

In this paper, we have considered the common adversarial model, where according to the adversary capabilities the following assumptions are made:

- The adversary $\mathcal{A}$ is having full control over the insecure public communication channel. $\mathcal{A}$ can intercept, modify, insert or delete any message.
- $\mathcal{A}$ knows the public keys and identities of all the participants.

Table 1. Notation Guide

p:	A large prime number ($p \geq 2^{160}$)
$E_p(a,b)$:	selected elliptic curve
Q:	A base point over $E_p(a,b)$
d_i:	Private key of the i^{th} legal user
$P_i = d_i \times P$:	Public key of the i^{th} legal user
M:	Message (plain text)
E_k/D_k:	Encryption/Decryption
T_i:	i^{th} Time stamp
$H(.)$:	A one way hash function
$\mathcal{U}_i$:	Legal user 'i'
$\mathcal{A}$:	Adversary
$\mathcal{M}$:	Merchant
$\mathcal{B}$:	Bank

3. Review of Yang et al.'s authenticated encryption scheme and 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 [36-38]. Further, it does not require digital signatures for verification. The scheme and its e-payment version are 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. The notation guide is illustrated in Table 1.

3.1.1. System initialization phase

This phase is committed to set system parameters. Initially a finite field F_p is selected over a large prime $p \geq 2^{160}$, then a non singular elliptic curve $E_p(a,b)$ is selected. Further system selects a base point Q over $E_p(a,b)$ along with a symmetric key encryption/decryption algorithm $E_k(.)/D_k(.)$. Each of the participants then picks his private d_i, and figures out his public key as $P_i = d_i \times Q$. Finally, system parameters and public keys are published, while private keys are kept secret by their corresponding users.

3.1.2. Authenticated encryption phase

This phase is executed when a legal user, say Alice, wants to send some confidential message to another legal user, say Bob. Alice initiates this phase as follows:

- Alice selects some random number $r \in Z_q$, then calculates $R = r \times P_a$.
- Alice then computes $\overline{R} = r \times P_b$ and $K = d_a \times \overline{R} = (k_x, k_y)$.
- By using k_x, Alice computes $C = E_{k_x}(ID_a \parallel m \parallel k_x \parallel T)$ and sends the tuple (C, R, T) to Bob.

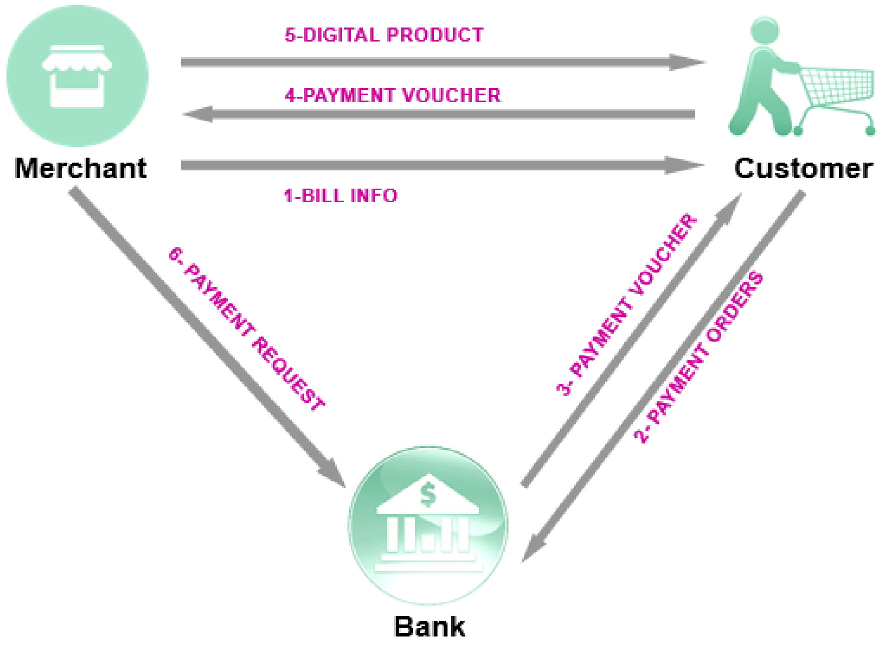
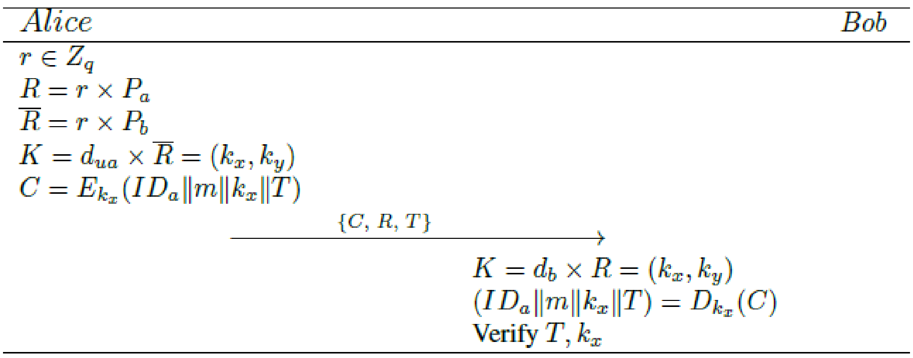


Figure 1. E-payment system


Alice *Bob*

$r \in Z_q$

$R = r \times P_a$

$\overline{R} = r \times P_b$

$K = d_{ua} \times \overline{R} = (k_x, k_y)$

$C = E_{k_x}(ID_a \| m \| k_x \| T)$

$\xrightarrow{\{C,\ R,\ T\}}$

$K = d_b \times R = (k_x, k_y)$

$(ID_a \| m \| k_x \| T) = D_{k_x}(C)$

Verify T, k_x

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

3.1.3. Verification phase

Upon receiving (C, R, T), Bob using his private key d_b verifies the received tuple as follows:

Step 1. Bob computes $K = d_b \times R = (k_x, k_y)$ to acquire the shared symmetric key k_x.

Step 2. Bob then decrypts C using k_x and obtains $(ID_a \parallel m \parallel k_x \parallel T)$ and verifies whether T is valid or not. If T is valid, then Bob verifies validity of k_x. If both T and k_x are valid, Bob considers the message is from legitimate user Alice.

The authenticated encryption and verification phases are also illustrated in Fig. 2.

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 customer Charlie, the merchant and the bank. Yang et al.'s scheme consists of the following five phases:

3.2.1. The initialization phase

During this phase the system parameters are initialized. The system sets $E_p(a, b)$, $E_k(.)$, $D_k(.)$ and base point Q. Then each participant, the customer, bank and the merchant, elects their respective key pairs P_C/d_C, P_B/d_B, P_M/d_M. Finally all public keys and system parameters are published.

3.2.2. Buying phase

The customer Charlie initiates this phase when he wants to buy some digitized product/s. Charlie visits merchant website and selects some product/s then he downloads goods/bill information $GI = goods_1, price_1, goods_1, price_2, \ldots good_k, price_k$. For buying digital product/s, Charlie performs the following steps:

Step 1. Charlie generates a random number $r \in Z_q$ and computes $R = r \times P_c$.

Step 2. Charlie then computes $\overline{R} = r \times P_B$ and $K = d_B \times \overline{R} = (k_x, k_y)$, where k_x, k_y are the respective x and y coordinates of K.

Step 3. Charlie then accumulates payment $p = \sum_{i=1}^{l} price_i$ and bill information $m = H(GI \parallel p \parallel ID_B)$

Step 4. Charlie computes $C_1 = E_{k_x}(ID_C \parallel m \parallel p \parallel k_x \parallel T_1)$, where T_1 is the current time stamp. Finally Charlie sends payment order tuple (C_1, R, T_1) to bank.

3.2.3. Paying phase

For the received payment order tuple (C_1, R, T_1), the bank performs the following steps to generate payment voucher and to check legality of the customer.

Step 1. The bank computes $K = d_B \times R = (k_x, k_y)$ to acquire the k_x.

Step 2. The bank then decrypts C_1 by using k_x and acquires $(ID_C \parallel m \parallel p \parallel k_x \parallel T_1) = D_{k_x}(C_1)$.

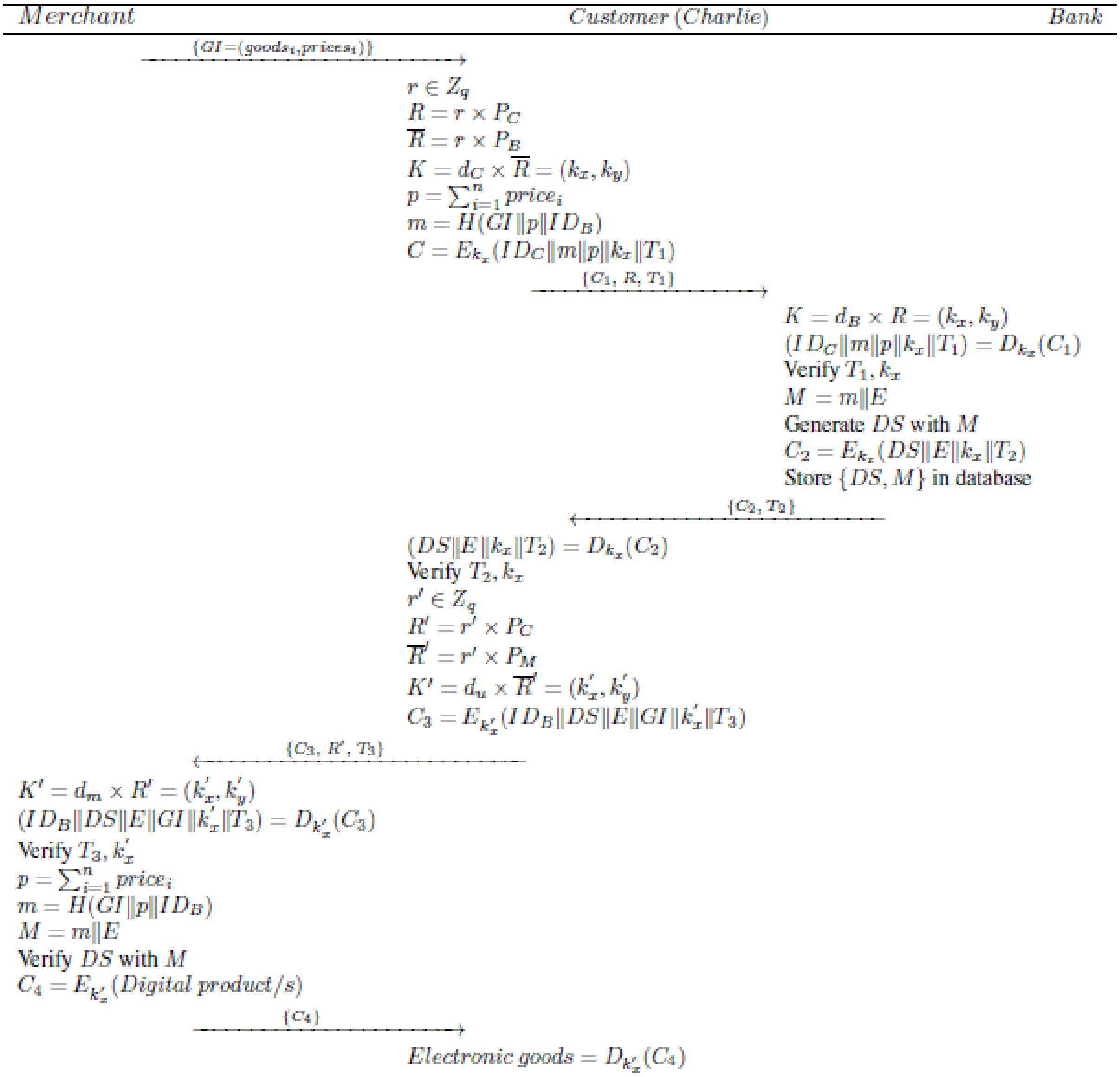

$Merchant$
$Customer\ (Charlie)$
$Bank$
$\{GI{=}(goods_i, prices_i)\}$
$r \in Z_q$
$R = r \times P_C$
$\overline{R} = r \times P_B$
$K = d_C \times \overline{R} = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI \| p \| ID_B)$
$C = E_{k_x}(ID_C \| m \| p \| k_x \| T_1)$
$\{C_1,\ R,\ T_1\}$
$K = d_B \times R = (k_x, k_y)$
$(ID_C \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m \| E$
Generate DS with M
$C_2 = E_{k_x}(DS \| E \| k_x \| T_2)$
Store $\{DS, M\}$ in database
$\{C_2, T_2\}$
$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$r' \in Z_q$
$R' = r' \times P_C$
$\overline{R}' = r' \times P_M$
$K' = d_u \times \overline{R}' = (k'_x, k'_y)$
$C_3 = E_{k'_x}(ID_B \| DS \| E \| GI \| k'_x \| T_3)$
$\{C_3,\ R',\ T_3\}$
$K' = d_m \times R' = (k'_x, k'_y)$
$(ID_B \| DS \| E \| GI \| k'_x \| T_3) = D_{k'_x}(C_3)$
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
$C_4 = E_{k'_x}(Digital\ product/s)$
$\{C_4\}$
$Electronic\ goods = D_{k'_x}(C_4)$

Figure 3. Yang el al‘s e-payment system

Step 3. The bank verifies whether T_1 and k_x are valid. If any of these is invalid, the bank aborts the session. Otherwise, the bank accepts the payment order.

Step 4. For valid received payment order, the bank withdraws amount p from Charlie's account and deposits p into a temporary account. Further, the bank generates an expiry date E and computes $M = m \parallel E$ along with digital signatures DS based on his private key d_B and message M . The bank archives the pair $\{DS, M\}$ in its database.

Step 5. Finally bank computes $C_2 = E_{k_x}(DS \parallel E \parallel k_x \parallel T_2)$ and sends payment voucher tuple (C_2, T_2) to Charlie.

Step 6. For the received payment voucher tuple (C_2, T_2) , Charlie using k_x decrypts C_2 to acquire $(DS \parallel E \parallel k_x \parallel T_2) = D_{k_x}(C_2)$. Charlie further checks validity of T_2 and k_x. If any of them in invalid, Charlie rejects the payment voucher. Otherwise he accepts the payment voucher.

3.2.4. Exchanging phase

For this phase, Charlie uses the valid payment voucher to exchange digital product/s with the merchant. To complete this phase, the following steps are performed between Charlie and merchant:

Step 1. Charlie generates a new random number $r' \in Z_q$ and computes $R' = r' \times P_C$, $\overline{R}' = r' \times P_M$ and $K' = d_C \times \overline{R}' = (k'_x, k'_y)$.

Step 2. Charlie then computes $C_3 = E_{k'_x}(ID_B \parallel DS \parallel E \parallel GI \parallel k'_x \parallel T_3)$ and sends payment voucher (C_3, R', T_3) to merchant.

Step 3. For the received payment voucher tuple (C_3, R', T_3) , the merchant computes $K' = d_M \times R' = (k'_x, k'_y)$ and $(ID_B \parallel DS \parallel E \parallel GI \parallel k'_x - \parallel T_3) = Dk'_x(C_3)$. Further merchant checks the validity of time stamp T_3 and k'_x and aborts the session if any of these is invalid. Otherwise, the merchant computes the bill information $p = \sum_{i=1}^{n} price_i$ and $m = H(GI \parallel p \parallel ID_B)$, $M = m \parallel E$. The merchant checks legality of signatures DS with M . If it is valid, merchant sends encrypted digital product/s $C_4 = E_{k'_x}(digital\ product/s)$ to Charlie. Upon receiving C_4, Charlie decrypts and gets the anticipated digital product/s.

3.2.5. Transferring phase

The merchant can send the payment voucher to bank before the expiry date. After expiry date the bank transfers amount p from temporary account to merchant's account and deletes the tuple $\{DS, M\}$ from his database.

4. Cryptanalysis of Yang et al.'s scheme

In this section, we show that Yang et al.'s authenticated encryption scheme can not withstand impersonation attacks, the same is the case with their e-payment system. In the following subsections, we prove that in both of Yang et al.'s schemes, an attacker can impersonate as a legal user having knowledge of only public keys and identities of the participants.

4.1. Impersonation attack on authenticated encryption

Here, we show that an adversary Eve can deceive the receiver Bob by performing the following steps:

Step 1. Eve computes the following:

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

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

Step 2. By using k_x Eve computes:

$$C = E_{k_x}(ID_a \parallel m \parallel k_x \parallel T) \tag{3}$$

Step 3. Eve finally, sends the authenticated encryption message tuple (C, R, T) to Bob.

Step 4. For the received tuple (C, R, T) , Bob computes the decryption key as follows:

$$K = d_b \times R = d_b \times Q = P_b = (k_x, k_y) \tag{4}$$

Step 5. Using k_x Bob computes:

$$(ID_a \parallel m \parallel k_x \parallel T) = D_{k_x}(C) \tag{5}$$

Step 6. Bob then verifies freshness of the time stamp T and validity of k_x. If both are valid, Bob recognizes the sender as Alice.

Proposition 1. At the end of the impersonation attack, Bob accepts Eve as the legal sender Alice.

Proof. During verification phase, the receiver Bob authenticates the sender Alice if time stamp T is fresh and T and k_x computed by Eq. (4) are same as of both received in encrypted message $C = E_{k_x}(ID_a \parallel m \parallel k_x \parallel T)$. Time stamp T can easily be generated by any adversary, so if the adversary Eve becomes able to generate valid k_x, then he can encrypt same key k_x and current time stamp along with arbitrary message and sender's ID_a. Therefore, the security merely relies on the validity of generation of k_x. As Eve computed R in Eq. (1) then Eve further computed K in Eq. (2) by just assigning public key P_b of Bob to K. Similarly,

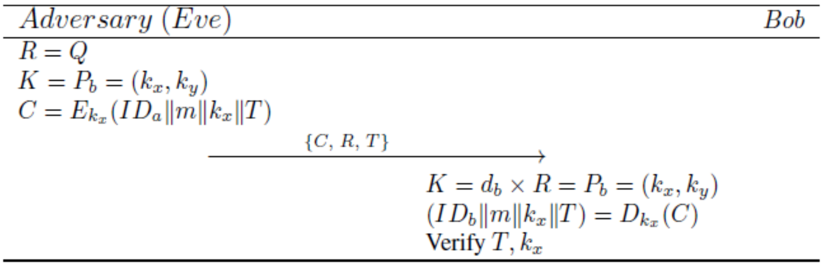

$Adversary\ (Eve)$
Bob
$R = Q$
$K = P_b = (k_x, k_y)$
$C = E_{k_x}(ID_a \| m \| k_x \| T)$
$\{C, R, T\}$
$K = d_b \times R = P_b = (k_x, k_y)$
$(ID_b \| m \| k_x \| T) = D_{k_x}(C)$
Verify T, k_x

Figure 4. Impersonation attack on Yang et al.‘s authenticated encryption scheme

upon receiving message, Bob computes same K in Eq. (4) as was computed by Eve in Eq. (2). So both Eq. (2) and (4) are same. Therefore, Eve successfully impersonated Bob on behalf of Alice.

4.2. Impersonation attack on e-payment system

Let Charlie be a legal customer and Eve be the adversary. Eve will perform the following steps in-order to masquerade as Charlie to deceive the bank and merchant for fraudulent purchase of electronic goods.

Step 1. Eve selects and downloads the goods information GI from merchant's website and computes following:

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

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

$$C = E_{k_x}(ID_C \parallel m \parallel k_x \parallel T_1) \tag{8}$$

Step 2. Eve sends $\{C_1,\ R,\ T_1\}$ to bank, where T_1 is the current time stamp.

Step 3. Upon receiving $\{C_1,\ R,\ T_1\}$, the bank computes the following:

$$K = d_B \times R = d_B \times Q = P_B = (k_x, k_y) \tag{9}$$

$$(ID_C \parallel m \parallel k_x \parallel T) = D_{k_x}(C_1) \tag{10}$$

Step 4. The bank verifies the correctness of T_1 and k_x after performing decryption. If both T_1 and k_x are correct, the bank generates the expiry date E and $M = m \parallel E$. Then it computes digital signature DS with M and computes:

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

Step 5. The bank deducts money from Charlie's account and stores $\{DS, M\}$ in his database. Finally, it sends $\{C_2, T_2\}$ to Charlie, where T_2 is fresh time stamp.

Step 6. Eve intercepts the message and uses the same key k_x to compute:

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

Step 7. Eve verifies T_2 and k_x, then computes the following:

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

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

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

Step 8. Eve sends $\{C_3, R', T_3\}$ to the merchant, where T_3 is fresh time stamp.

Step 9. Upon receiving $\{C_3, R', T_3\}$, the merchant computes the following:

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

$$(ID_B \parallel DS \parallel E \parallel GI \parallel k_x' \parallel T_3) = D_{k_x'}(C_3) \tag{17}$$

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

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

$$m = H(GI \parallel p \parallel ID_B) \tag{19}$$

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

Step 11. Further merchant computes digital signature DS based on M and checks its validity by comparing it to the DS obtained in Eq. (17). If it is valid, the merchant computes:

$$C_4 = E_{k_x'}(Digital\ product/s) \tag{21}$$

Step 12. Finally, the merchant sends encrypted electronic goods C_4 to Charlie.

Step 13. Eve intercepts the message and retrieves $Digital\ product/s = D_{k_x'}(C_4)$.

Proposition 2. At the end of the impersonation attack, The Bank and the merchant accept adversary Eve as the legal customer Charlie and Eve becomes able to purchase digital product/s on behalf of Charlie.

Proof. During paying phase, the bank authenticates the customer Charlie on the basis of time stamp T_1 and k_x. Following are the three conditions to be met for successful impersonation attack:

1. T_1 is fresh.
2. T_1 should be the same as received in plain text and obtained after decryption of C_1.
3. k_x should be the same as computed in Eq. (9) and obtained after decryption of C_1.

The fresh time stamp T_1 can be easily generated by any adversary, so the security of Yang et al.'s scheme relies on the validity of k_x. The adversary Eve computed $R = Q$ in Eq. (6) and $K = P_B = (k_x, k_y)$ in Eq. (7), then Eve generated fresh time stamp T_1. Eve further sent (C_1, R, T_1) to the bank, which initially verified freshness of T_1. T_1 was freshly generated time stamp so the bank computed $K = d_B \times R = d_B \times Q = P_B = (k_x, k_y)$ in Eq. (9). Then by using k_x, bank further computed $(ID_C \parallel m \parallel k_x \parallel T) = D_{k_x}(C_1)$ in Eq. (10) which is same as computed by Eve in Eq. (7), and found same in C_1 in Eq. (10). Similarly T_1 is also same as it is sent in plain text. Therefore, Eve has deceived the bank on behalf of Charlie. Similarly, during exchanging phase Eve computed $R' = Q$ in Eq. (13) and $K' = P_M$ in Eq. (14). Eve further generated new time stamp T_3 and sent (C_3, R', T_3) to the merchant. The same three conditions must be met for successful impersonation attack, which are as follows:

1. T_3 is fresh.
2. T_3 should be the same as received in plain text and obtained after decryption of C_3.
3. $k_x^{'}$ should be the same as computed in Eq. (16) and obtained after decryption of C_3.

As described earlier generation of fresh time stamp is very easy, the security relies on the validity of $k_x^{'}$. The merchant upon receiving first checks whether T_3 is fresh or not, as T_3 was freshly generated, so merchant computes $K' = d_M \times R' = (k_x', k_y')$ in Eq. (16) and then decrypts C_3, which is same as the merchant computed K' because it was computed by Eve in Eq. (14). Hence the merchant finds same $k_x^{'}$ after decryption of C_3. Therefore, Eve has impersonated the bank and the merchant on behalf of Charlie and fraudulently purchased digital product/s.

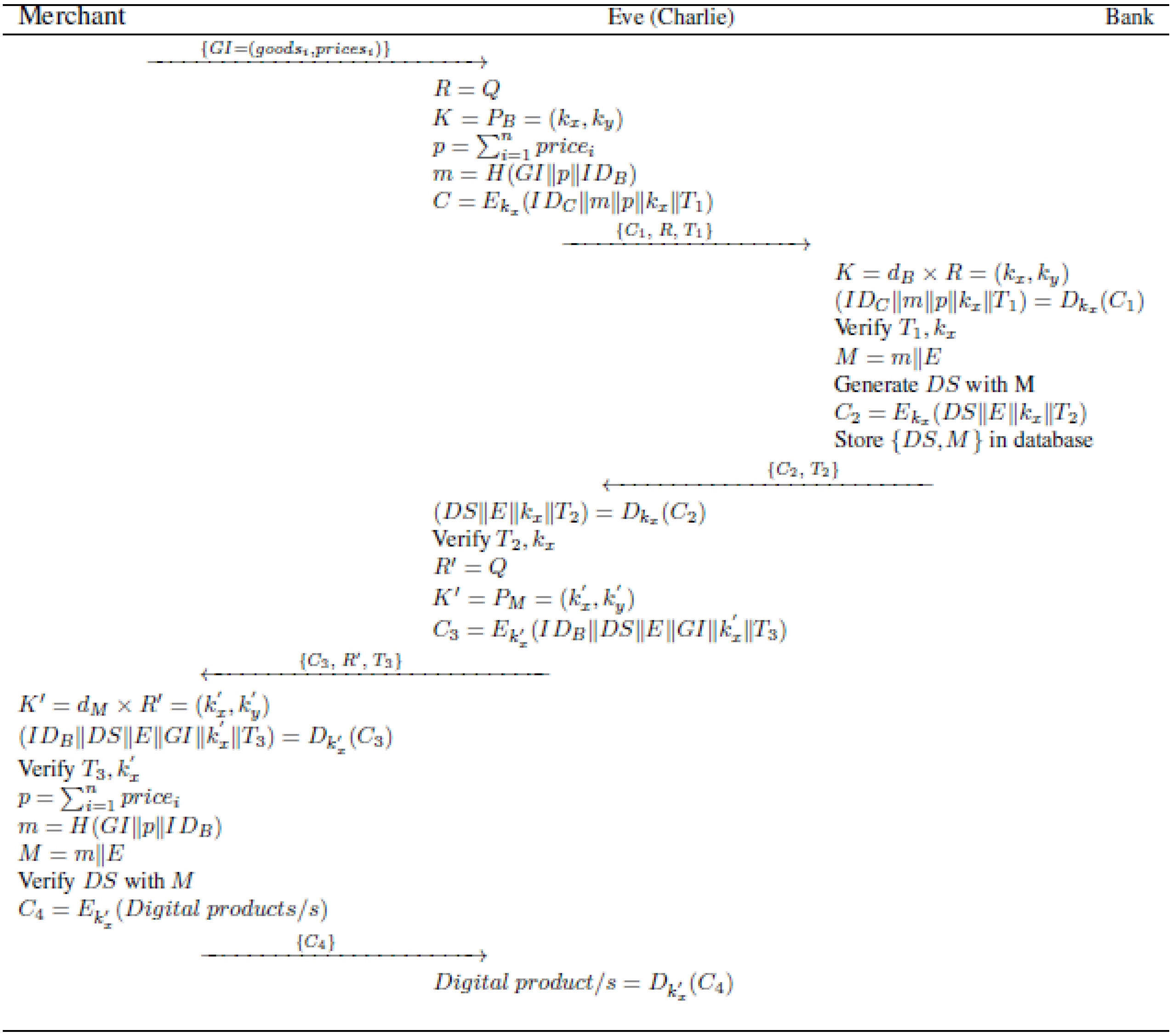

Merchant
Eve (Charlie)
Bank
$\{GI=(goods_i, prices_i)\}$
$R = Q$
$K = P_B = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI \| p \| ID_B)$
$C = E_{k_x}(ID_C \| m \| p \| k_x \| T_1)$
$\{C_1, R, T_1\}$
$K = d_B \times R = (k_x, k_y)$
$(ID_C \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m \| E$
Generate DS with M
$C_2 = E_{k_x}(DS \| E \| k_x \| T_2)$
Store $\{DS, M\}$ in database
$\{C_2, T_2\}$
$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$R' = Q$
$K' = P_M = (k_x', k_y')$
$C_3 = E_{k_x'}(ID_B \| DS \| E \| GI \| k_x' \| T_3)$
$\{C_3, R', T_3\}$
$K' = d_M \times R' = (k_x', k_y')$
$(ID_B \| DS \| E \| GI \| k_x' \| T_3) = D_{k_x'}(C_3)$
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
$C_4 = E_{k_x'}(Digital\ products/s)$
$\{C_4\}$
$Digital\ product/s = D_{k_x'}(C_4)$

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

5. 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 the initialization phase. Proposed authenticated encryption scheme is shown in Fig. 6.

5.1. Authenticated encryption phase

Authenticated encryption is performed by a legal user Alice, when she wants to send another user Bob a message m. Alice performs the following steps:

Step 1. Alice generates a random number $r \in Z_p$ and uses her private key d_a along with fresh time stamp T to compute $R = (d_a + T)/r$.

Step 2. Alice then figures out $K = r \times P_b$ and extracts k_x (the x coordinate of K).

Step 3. Using k_x Alice computes $C = E_{k_x}(ID_a \parallel m \parallel k_x \parallel T)$, and sends (C, R, T) to Bob.

5.2. Verification phase

For the received tuple (C, R, T), Bob performs following steps to acquire message and verify the legality of sender Alice. For verification Bob performs the following steps:

Step 1. Bob computes $K = R(P_a + TQ)d_b = (k_x, k_y)$ to obtain key k_x.

Step 2. Bob using k_x, decrypts C and gets $(ID_a \parallel M \parallel k_x \parallel T)$,

Step 3. Bob verifies whether the received T and computed k_x are the same as they are present in decrypted message. if both are same then he considers the sender as legal user Alice.

6. The improved e-payment using proposed scheme

In this section, we elaborate the enhanced e-payment system, which is based on our proposed authenticated encryption scheme. The enhanced e-payment system is illustrated in Fig. 7 and is described by following subsections:

6.1. Initialization phase

This phase is analogous to that presented in Subsection 3.2.1. In this phase the system parameters are initialized. The system sets $E_p(a, b)$, $E_k(.)$, $D_k(.)$ and base point Q. Then each participant, the customer, bank and the merchant, elects their respective key pairs P_C/d_C, P_B/d_B, P_M/d_M. Finally all public keys and system parameters are published.

6.2. Buying Phase

The customer Charlie initiates this phase when he wants to buy some digitized product/s. Charlie visits merchant website and selects some product/s. Then he downloads goods/bill information GI. For buying digital product/s, Charlie performs the following steps:

Step 1. Charlie generates a random number $r \in Z_q$ and computes $R = (d_C + T_1)/r$.

Step 2. Charlie then computes $K = r \times P_B = (k_x, k_y)$, where k_x, k_y are the respective x and y coordinates of K.

Step 3. Charlie then accumulates payment $p = \sum_{i=1}^{n} price_i$ and bill information $m = H(GI \parallel p \parallel ID_B)$.

Step 4. Charlie computes $C_1 = E_{k_x}(ID_C \parallel m \parallel p \parallel k_x \parallel T_1)$, where T_1 is the current time stamp. Finally Charlie sends payment order tuple (C_1, R, T_1) to the bank.

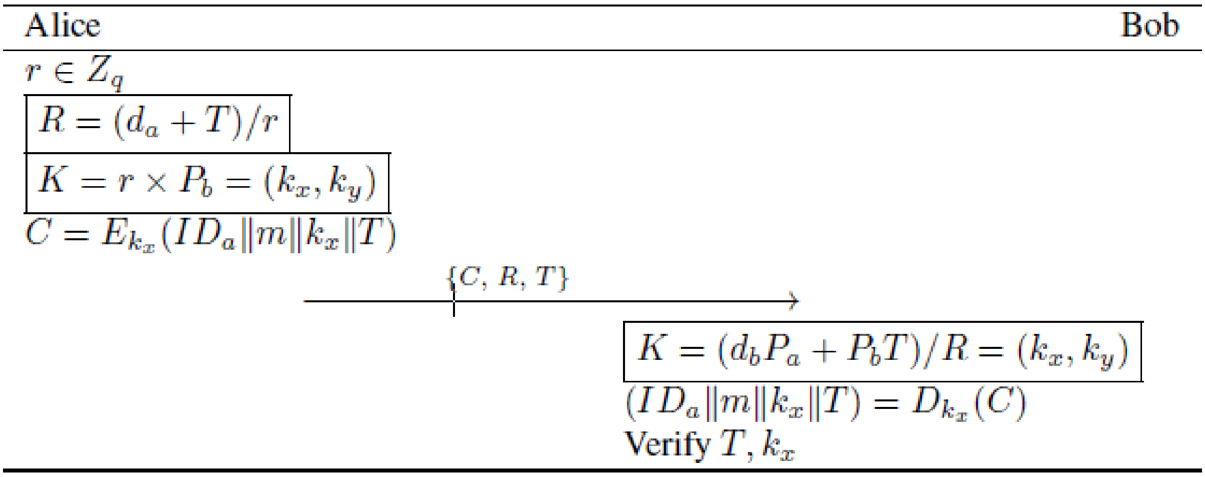

Alice
Bob
$r \in Z_q$
$R = (d_a + T)/r$
$K = r \times P_b = (k_x, k_y)$
$C = E_{k_x}(ID_a \| m \| k_x \| T)$
$\{C, R, T\}$
$K = (d_b P_a + P_b T)/R = (k_x, k_y)$
$(ID_a \| m \| k_x \| T) = D_{k_x}(C)$
Verify T, k_x

Figure 6. Proposed authenticated encryption scheme

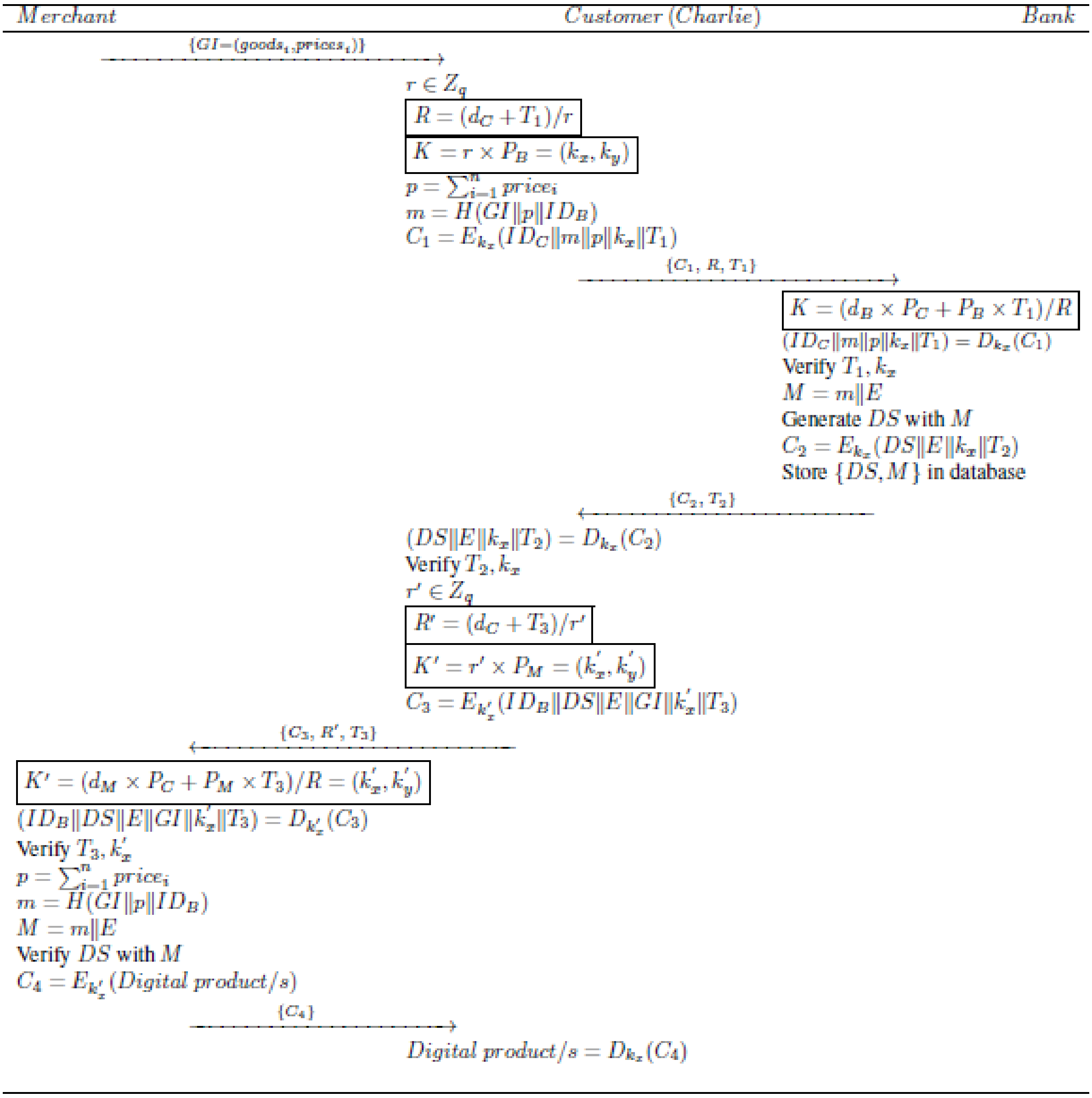

Merchant
Customer (Charlie)
Bank
{GI=(goods_i, prices_i)}
$r \in Z_q$
$R = (d_C + T_1)/r$
$K = r \times P_B = (k_x, k_y)$
$p = \sum_{i=1}^{n} price_i$
$m = H(GI \| p \| ID_B)$
$C_1 = E_{k_x}(ID_C \| m \| p \| k_x \| T_1)$
$\{C_1, R, T_1\}$
$K = (d_B \times P_C + P_B \times T_1)/R$
$(ID_C \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$
Verify T_1, k_x
$M = m \| E$
Generate DS with M
$C_2 = E_{k_x}(DS \| E \| k_x \| T_2)$
Store $\{DS, M\}$ in database
$\{C_2, T_2\}$
$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2)$
Verify T_2, k_x
$r' \in Z_q$
$R' = (d_C + T_3)/r'$
$K' = r' \times P_M = (k_x', k_y')$
$C_3 = E_{k_x'}(ID_B \| DS \| E \| GI \| k_x' \| T_3)$
$\{C_3, R', T_3\}$
$K' = (d_M \times P_C + P_M \times T_3)/R = (k_x', k_y')$
$(ID_B \| DS \| E \| GI \| k_x' \| T_3) = D_{k_x'}(C_3)$
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
$C_4 = E_{k_x'}(Digital\ product/s)$
$\{C_4\}$
$Digital\ product/s = D_{k_x}(C_4)$

Figure 7. Proposed e-payment system

6.3. Paying phase

For the received payment order tuple $\{C_1,\ R,\ T_1\}$, the bank performs the following steps:

Step 1. The bank computes $K = (d_B \times P_C + P_B \times T_1)/R = (k_x, k_y)$ to acquire the k_x.

Step 2. The bank then decrypts C_1 by using k_x and acquires $(ID_C \parallel m \parallel p \parallel k_x \parallel T_1) = D_{k_x}(C_1)$.

Step 3. The Bank verifies whether T_1 and k_x are valid. If any of these is invalid , the bank aborts the session. Otherwise, the bank accepts the payment order.

Step 4. For valid received payment order, the bank withdraws amount p from Charlie's account and deposits p into a temporary account. Further, the bank generates an expiry date E and computes $M = m \parallel E$ along with digital signatures DS based on it's private key d_B and message M. The bank archives the pair $\{DS, M\}$ in its database.

Step 5. Finally, the bank computes $C_2 = E_{k_x}(DS \parallel E \parallel k_x \parallel T_2)$ and sends payment voucher pair (C_2, T_2) to Charlie.

Step 6. For the received payment voucher pair (C_2, T_2) , Charlie using k_x decrypts C_2 to acquire $(DS \parallel E \parallel k_x \parallel T_2) = D_{k_x}(C_2)$. Charlie further checks the validity of T_2 and k_x. If any of these in invalid, Charlie rejects the payment voucher; otherwise he accepts the payment voucher.

6.4. Exchange phase

The exchange phase consists of the following three steps:

Step 1. For the valid payment voucher, Charlie generates $r' \in Z_p$ to compute $R' = (d_C + T_3)/r'$ and $K' = r' \times P_M = (k'_x, k'_y)$. Then, using $K_x^{'}$, Charlie computes $C_3 = E_{k'_x}(ID_B \parallel DS \parallel E \parallel GI \parallel k_x^{'} \parallel T_3)$. Finally, he sends (C_3, R', T_3) to merchant.

Step 2. Upon Receiving (C_3, R', T_3), the merchant computes $K' = (d_M \times P_C + P_M \times T_3)/R = (k'_x, k'_y)$, then decrypts C_3, using $k_x^{'}$ as decryption key. Further merchant checks the validity of time stamp T_3 and $k_x^{'}$ and aborts the session if any of these are invalid. Otherwise, merchant computes the bill information $p = \sum_{i=1}^{n} price_i$ and $m = H(GI \parallel p \parallel ID_B)$, $M = m \parallel E$. The merchant checks legality of signatures DS with M. If it is valid, the merchant sends encrypted digital product/s $C_4 = E_{k'_x}(digital\ product/s)$ to Charlie.

Step 3. Upon receiving C_4, Charlie decrypts and gets the anticipated digital product/s.

6.5. Transferring phase

In the proposed e-payment system the merchant can send the payment voucher to the bank before its expiry date, the customer can abandon the transaction before the expiry date, if some dispute arise between the merchant and himself. In such case the dispute is referred to the trusted third party. Otherwise, after expiry date the bank transfers the transaction amount to merchant's account and obliterate the record $\{DS, M\}$ from his database.

6.6. Dispute resolution phase

A trusted third party (TTP) can resolve a dispute if it arises between the merchant and Charlie (the customer). For dispute resolution, private key of the merchant is exposed to trusted third party, which can check the validity of payment voucher and encryption key $k_x^{'}$ after acquiring the message $\{C_3, R', T_3\}$. TTP performs the following for fixing responsibility:

$$K' = (d_M \times P_C + P_M \times T_3)/R_3 = (k'_x, k'_y) \quad (22)$$

$$(ID_B \parallel DS \parallel E \parallel GI \parallel k'_x \parallel T_3) = D_{k'_x}(C_3) \quad (23)$$

After decrypting C_3, TTP checks the equality of T_3 with in C_3 and received in plain text message. Further, TTP checks whether $k_x^{'}$, received in encrypted message (as computed in Eq.(23)) is same as used during decryption of C_3 (as computed in Eq. (22)).

7. Security analysis

The proposed authenticated encryption scheme and e-payment system fulfill all the known security requirements as stated by Yang et al. and others. In this section, we analyze the security of the proposed schemes. The proposed schemes provide resistance to replay, outsider, impersonation, man-in-middle and ID theft attacks. Further, it provides confidentiality, authentication, privacy protection and non-repudiation. Additionally our proposed e-payment system also provides double spending prevention of the same payment voucher. Table 8 illustrates the security comparison of the proposed scheme with Yang et al.'s scheme.

7.1. Mutual Authentication

In the proposed authenticated encryption scheme, Alice (the sender) generates a symmetric key k_x. The computation of k_x involves: (1) engendering a random number $r \in Z_p$; (2) computing $R = (d_a + T)/r$; and (3) $K = r \times P_b = (k_x, k_y)$. As it can be easily seen, that valid pair (K, R) can only be generated by making use of Alice's private key d_a and Bob's (the receiver) pubic key P_b. Similarly, on receiving side, Bob can generate the same symmetric key k_x using his own private key d_b and Alice's public key P_a. Hence the shared symmetric key k_x can only be computed and verified by legal entities in authenticated encryption scheme. E-payment system is very analogous to authenticated encryption scheme. Hence both the proposed schemes ensure mutual authentication.

7.2. Integrity

If some adversary modifies the authenticated encryption message (C, R, T), then Bob will reject the message, as the message will not pass the validity check for k_x and T.

7.3. Privacy protection

In the proposed schemes, the identity of sender is sent in encrypted message C. Further, in e-payment system, the goods information sent to bank is protected by one-way hash function. It must also be noted that the digital signature does not expose the customer's information. Hence sender as well as buying information privacy is protected.

7.4. Non-repudiation

In proposed schemes, a trusted third party can verify the transaction and legality of sender and receiver as illustrated in subsection 6.6. Hence none of the participant can repudiate its part in the transaction.

7.5. Impersonation attack

An adversary (Eve) can impersonate as a legal user/customer (Alice) to deceive the receiver (Bob), if he is having the ability to generate valid (R, K) pair. It is already described in Subsection 7.1, that the adversary needs Alice's private key d_a and Bob's public key P_b to generate valid (R, K) pair. Similarly, the adversary needs to know Bob's private key d_b and Alice's public key P_a, if he wants to impersonates as the receiver (Bob). However, the adversary cannot access private keys, so our scheme is resistant to impersonation attacks.

7.6. Replay attack

The proposed scheme resists replay attacks, as if some adversary intercepts a previous message (C, R, T) sent by Alice and replays it later to Bob. Then upon reception of the message tuple, Bob first checks the validity of time stamp T, because the time stamp is outdated, Bob recognizes the replay and simply rejects the message.

7.7. Outsider attack

An outsider can easily intercept a message (C, R, T). Further, he can also access public keys and identities of the participants, but to attain the useful information contained in C, he needs to know private key of Alice (the sender).

7.8. Man-in-middle attack

If the adversary (Eve) intercepts a message (C, R, T) and replaces it by some other message tuple $(\bar{C}, \bar{R}, T_{fre})$, then the receiver (Bob) can easily understand the message is from adversary, as the valid $(\bar{C}, \bar{R})$ can be generated by the use of private key of Alice. Suppose if adversary sends the same intercepted tuple (C, R) along with fresh time stamp T_{fre} , then Bob after decryption of message will compare the time stamp embedded in the encrypted message C and plain text time stamp T_{fre}, because both are not the same, Bob will reject the message.

7.9. ID theft attack

The valid pair (C, R) is generated using public and private keys of receiver and sender, respectively. Furthermore, (C, R) pair is having no relationship with the identities of the participants. Therefore, even if identities of all the participants are exposed to adversary, it will have no effect on security of the proposed schemes.

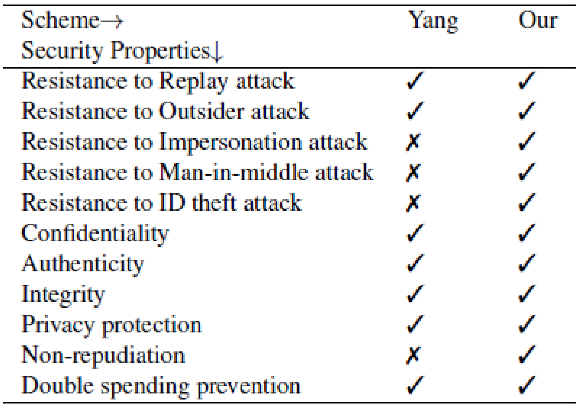
7.10. Confidentiality

The encrypted message C can only be decrypted by first computing the shared symmetric key k_x. It is already proved in Subsection 7.1, that only legal intended user Bob can compute the symmetric key k_x. Hence, proposed scheme provides message confidentiality.

7.11. Double spending prevention

In the proposed e-payment system the payment voucher $\{DS, M\}$ is used once as the voucher remains in bank's database until the merchant asks for payment. After expiry date bank transfers the voucher amount in merchant's account and deletes the voucher from his database. Hence, our e-payment system ensures double spending prevention of same voucher.

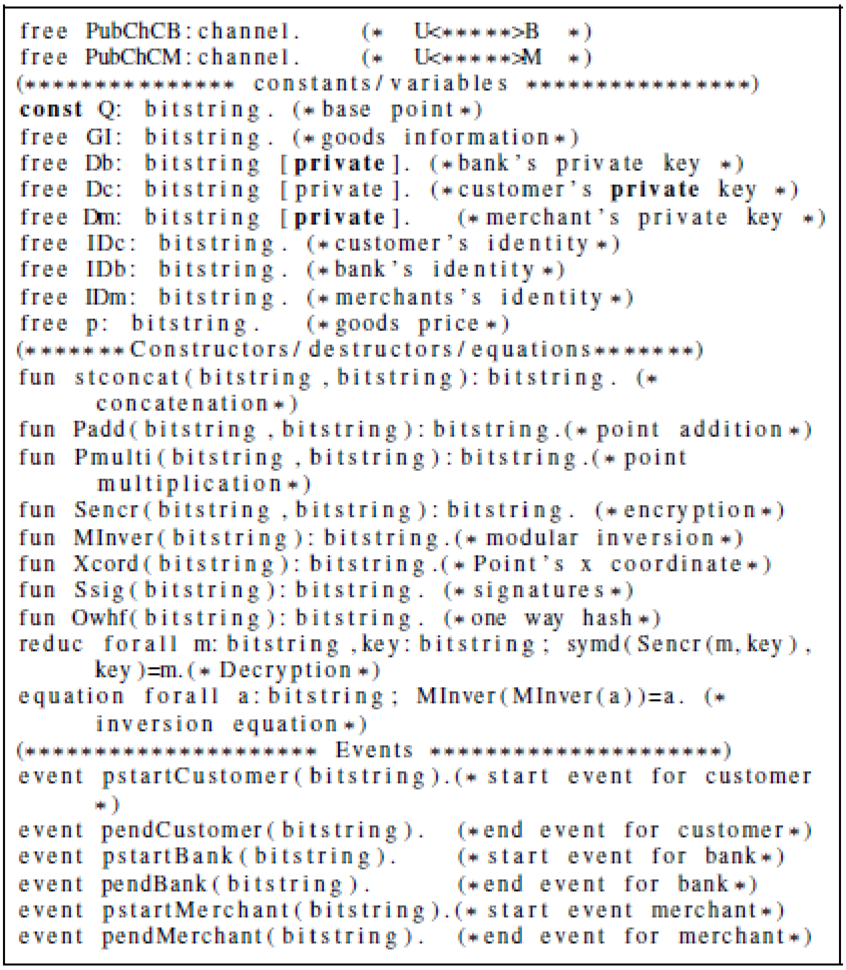
Table 2. E-payment system security analysis



Scheme→ Security Properties↓	Yang	Our
Resistance to Replay attack	✓	✓
Resistance to Outsider attack	✓	✓
Resistance to Impersonation attack	✗	✓
Resistance to Man-in-middle attack	✗	✓
Resistance to ID theft attack	✗	✓
Confidentiality	✓	✓
Authenticity	✓	✓
Integrity	✓	✓
Privacy protection	✓	✓
Non-repudiation	✗	✓
Double spending prevention	✓	✓

8. Protocol verification using ProVerif

To substantiate the security of proposed e-payment scheme, we have adopted formal ProVerif model. Formal security analysis for cryptographic protocols was initiated during mid 80's with varying techniques including algebraic, state space and logic methods. Applied pi calculus is one of the prevailing logic methods for formal analysis of cryptographic protocols. ProVerif makes use of applied pi calculus to validate correctness and robustness of security protocols [39]. The analysis capabilities of ProVerif ranges from proving the trace properties like authentication, reachability and secrecy to ascertain whether or not a presented protocol extends to a bad state [40], to the observational properties like anonymity and privacy [41, 42]. ProVerif protocol model consists of three parts. In declaration part, names and cryptographic primitives are stated. In process parts, the processes and the subprocesses are defined, while core protocol steps are defined in the main part. We have simulated the proposed protocol steps as described in Subsection 6, also illustrated in Fig. 7. The declaration part consists of modeling of two public channels along with names, constructors, destructors and events purely defined as of the identities, keys, and cryptographic primitives used in the proposed protocol. Fig. 1 illustrates the declaration phase. Fig. 2 shows three distinct processes, the customer, the bank and the merchant. All the processes

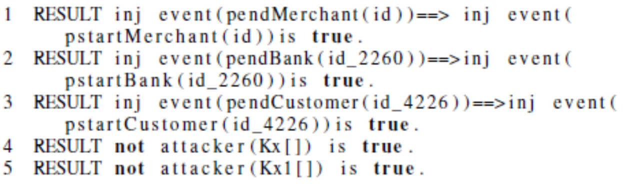


```
free PubChCB:channel.     (*  U<*****>B  *)
free PubChCM:channel.     (*  U<*****>M  *)
(*************** constants/variables ***************)
const Q: bitstring. (*base point*)
free GI: bitstring. (*goods information*)
free Db: bitstring [private]. (*bank's private key *)
free Dc: bitstring [private]. (*customer's private key *)
free Dm: bitstring [private].   (*merchant's private key *)
free IDc: bitstring. (*customer's identity*)
free IDb: bitstring. (*bank's identity*)
free IDm: bitstring. (*merchants's identity*)
free p: bitstring.    (*goods price*)
(*******Constructors/destructors/equations*******)
fun stconcat(bitstring,bitstring):bitstring. (*
      concatenation*)
fun Padd(bitstring,bitstring):bitstring.(*point addition*)
fun Pmulti(bitstring,bitstring):bitstring.(*point
      multiplication*)
fun Sencr(bitstring,bitstring):bitstring. (*encryption*)
fun MInver(bitstring):bitstring.(*modular inversion*)
fun Xcord(bitstring):bitstring.(*Point's x coordinate*)
fun Ssig(bitstring):bitstring. (*signatures*)
fun Owhf(bitstring):bitstring. (*one way hash*)
reduc forall m:bitstring,key:bitstring; symd(Sencr(m,key),
      key)=m.(*Decryption*)
equation forall a:bitstring; MInver(MInver(a))=a. (*
      inversion equation*)
(********************** Events *********************)
event pstartCustomer(bitstring).(*start event for customer
      *)
event pendCustomer(bitstring).  (*end event for customer*)
event pstartBank(bitstring).    (*start event for bank*)
event pendBank(bitstring).      (*end event for bank*)
event pstartMerchant(bitstring).(*start event merchant*)
event pendMerchant(bitstring).  (*end event for merchant*)
```

Figure 8. Declaration part

are modeled as the real steps in proposed protocol. The customer process initiates the transaction by computing and sending ***(C1,R,T1)*** to the bank via public channel ***PubChCB***. The bank process checks legality of the sender and message and then responds with ***(C2,T2)*** via same channel. The customer process then sends ***(C3,R',T3)*** to merchant process, which after verification of authenticity responds with ***(C4)*** the digital product/s. We simulate parallel execution of all the three processes in the main part along with the attacker queries to check secrecy of two session keys shared between the customer and the bank and the customer and the merchant. Further, we also model the reachability queries to check correctness of the proposed schemes. The queries are shown in Fig. 10.

Finally, we got the results as follows:



```
1  RESULT inj event(pendMerchant(id))==> inj event(
      pstartMerchant(id))is true.
2  RESULT inj event(pendBank(id_2260))==>inj event(
      pstartBank(id_2260))is true.
3  RESULT inj event(pendCustomer(id_4226))==>inj event(
      pstartCustomer(id_4226))is true.
4  RESULT not attacker(Kx[]) is true.
5  RESULT not attacker(Kx1[]) is true.
```

The results $(1-3)$ state the successful initiation and termination for three process. While $(4-5)$ verify that attacker is not able to reveal two session key Kx and $Kx1$, which eventually proves that the proposed scheme achieved mutual authentication between the customer and the bank as well as between the customer and the merchant.

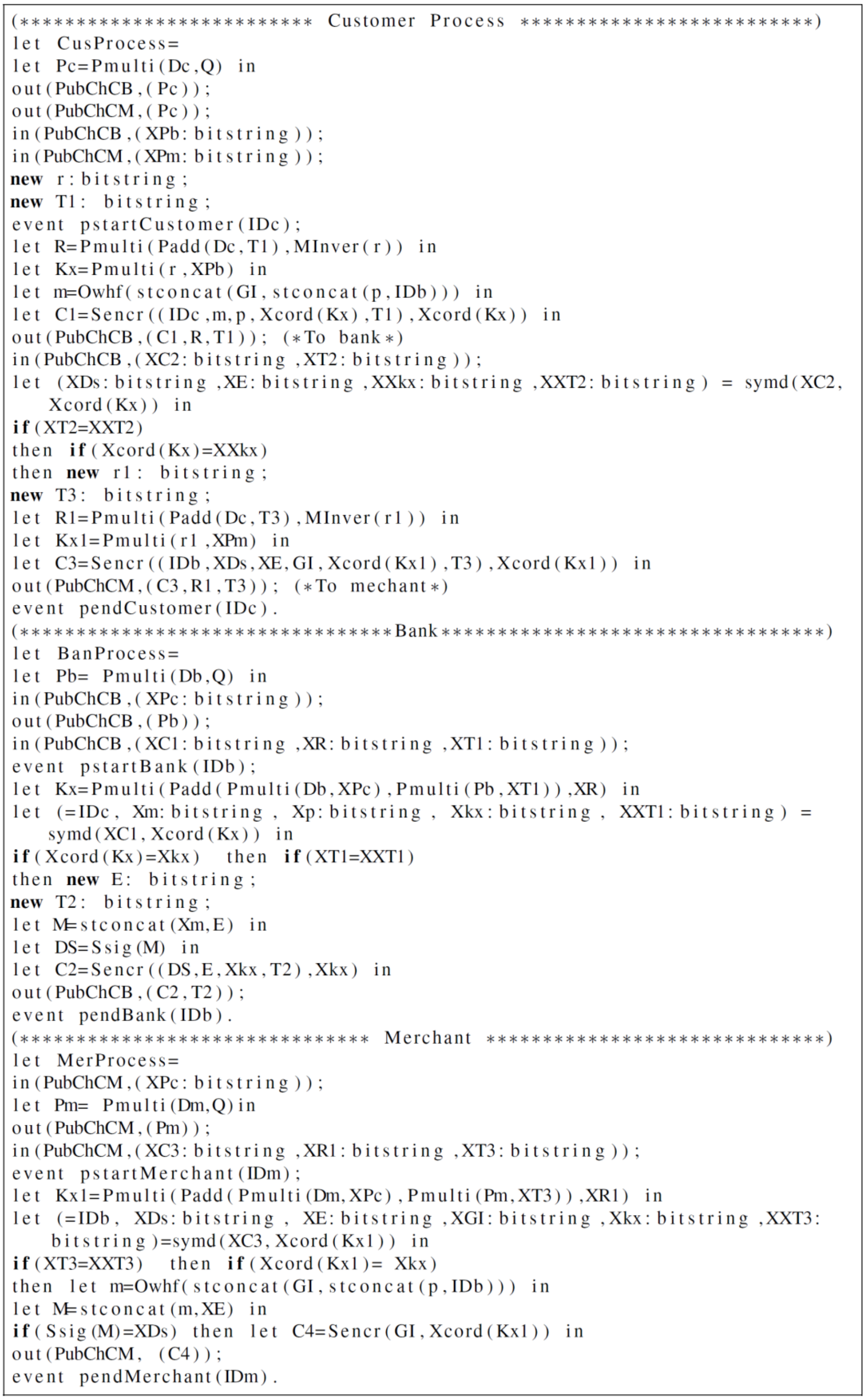
9. Performance Analysis

This section evaluates the performance comparison of the proposed e-payment scheme with related existing schemes [33, 43, 44]. For comparison purposes, we have introduced the following notations:

- T_{me}: Time for Exponentiation
- T_{ecm}: Time for Elliptic curve point multiplication
- T_{eca}: Time for Elliptic curve point addition
- T_{syed}: Time for Symmetric encryption/ decryption operation
- T_{hs}: Time for One way hash function

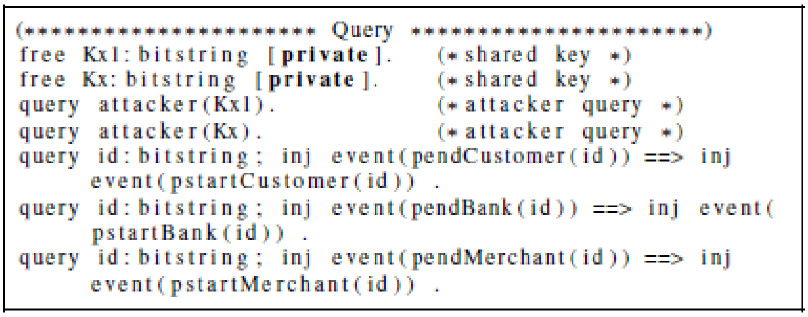
Table 8 illustrates the performance comparison of the proposed scheme with related existing schemes [33, 43, 44]. According to Kilinc et al. [45], execution time for $T_{me} \approx 3.85\ ms$, $T_{ecm} \approx 2.226\ ms$, $T_{eca} \approx 0.0288\ ms$, $T_{syed} \approx 0.0046\ ms$ and $T_{hs} \approx$

0.0023 ms on Ubuntu system with dual core E2200 2.20 GHz processor, 2048 MB of RAM and PBC library. It can be easily verified that the execution time of the proposed scheme is just 9% of Wang et al.'s scheme [44], 37% of Oros-Popescu's scheme [43] and 50% of Yang et al.'s scheme [33]. Therefore, the proposed scheme is having more security and is more lightweight as compared with existing schemes.



```
(************************** Customer Process **************************)
let CusProcess=
let Pc=Pmulti(Dc,Q) in
out(PubChCB,(Pc));
out(PubChCM,(Pc));
in(PubChCB,(XPb:bitstring));
in(PubChCM,(XPm:bitstring));
new r:bitstring;
new T1: bitstring;
event pstartCustomer(IDc);
let R=Pmulti(Padd(Dc,T1),MInver(r)) in
let Kx=Pmulti(r,XPb) in
let m=Owhf(stconcat(GI,stconcat(p,IDb))) in
let C1=Sencr((IDc,m,p,Xcord(Kx),T1),Xcord(Kx)) in
out(PubChCB,(C1,R,T1)); (*To bank*)
in(PubChCB,(XC2:bitstring,XT2:bitstring));
let (XDs:bitstring,XE:bitstring,XXkx:bitstring,XXT2:bitstring) = symd(XC2,
    Xcord(Kx)) in
if(XT2=XXT2)
then if(Xcord(Kx)=XXkx)
then new r1: bitstring;
new T3: bitstring;
let R1=Pmulti(Padd(Dc,T3),MInver(r1)) in
let Kx1=Pmulti(r1,XPm) in
let C3=Sencr((IDb,XDs,XE,GI,Xcord(Kx1),T3),Xcord(Kx1)) in
out(PubChCM,(C3,R1,T3)); (*To mechant*)
event pendCustomer(IDc).
(********************************Bank********************************)
let BanProcess=
let Pb= Pmulti(Db,Q) in
in(PubChCB,(XPc:bitstring));
out(PubChCB,(Pb));
in(PubChCB,(XC1:bitstring,XR:bitstring,XT1:bitstring));
event pstartBank(IDb);
let Kx=Pmulti(Padd(Pmulti(Db,XPc),Pmulti(Pb,XT1)),XR) in
let (=IDc, Xm:bitstring, Xp:bitstring, Xkx:bitstring, XXT1:bitstring) =
    symd(XC1,Xcord(Kx)) in
if(Xcord(Kx)=Xkx) then if(XT1=XXT1)
then new E: bitstring;
new T2: bitstring;
let M=stconcat(Xm,E) in
let DS=Ssig(M) in
let C2=Sencr((DS,E,Xkx,T2),Xkx) in
out(PubChCB,(C2,T2));
event pendBank(IDb).
(****************************** Merchant ******************************)
let MerProcess=
in(PubChCM,(XPc:bitstring));
let Pm= Pmulti(Dm,Q)in
out(PubChCM,(Pm));
in(PubChCM,(XC3:bitstring,XR1:bitstring,XT3:bitstring));
event pstartMerchant(IDm);
let Kx1=Pmulti(Padd(Pmulti(Dm,XPc),Pmulti(Pm,XT3)),XR1) in
let (=IDb, XDs:bitstring, XE:bitstring,XGI:bitstring,Xkx:bitstring,XXT3:
    bitstring)=symd(XC3,Xcord(Kx1)) in
if(XT3=XXT3) then if(Xcord(Kx1)= Xkx)
then let m=Owhf(stconcat(GI,stconcat(p,IDb))) in
let M=stconcat(m,XE) in
if(Ssig(M)=XDs) then let C4=Sencr(GI,Xcord(Kx1)) in
out(PubChCM, (C4));
event pendMerchant(IDm).
```

Figure 9. Process part



```
(*********************** Query ***********************)
free Kx1: bitstring [private].    (*shared key *)
free Kx: bitstring [private].     (*shared key *)
query attacker(Kx1).              (*attacker query *)
query attacker(Kx).               (*attacker query *)
query id: bitstring; inj event(pendCustomer(id)) ==> inj
      event(pstartCustomer(id)) .
query id: bitstring; inj event(pendBank(id)) ==> inj event(
      pstartBank(id)) .
query id: bitstring; inj event(pendMerchant(id)) ==> inj
      event(pstartMerchant(id)) .
```

Figure 10. Main part

10. 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 a 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 the proposed scheme improved the security as well as computation overhead and is more suitable for resource constrained environments.

Table 3. Computation cost analysis

Scheme→ Phase↓	Wang et al. [44]	Oros-Popescu's [43]	Yang et al. [33]	Proposed
Buying Phase	$9T_{me}$	$8T_{ecm}$	$3T_{ecm} + 1T_{syed}$	$1T_{ecm} + 1T_{syed}$
Paying Phase	$11T_{me}$	$2T_{ecm} + 3T_{eca}$	$1T_{ecm} + 2T_{syed}$	$1T_{ecm} + 2T_{syed}$
Exchanging Phase	$5T_{me}$	$3T_{eca}$	$4T_{ecm} + 2T_{syed}$	$2T_{ecm} + 5T_{syed}$
Total	$25T_{me}$	$10T_{ecm} + 6T_{eca}$	$8T_{ecm} + 5T_{syed}$	$4T_{ecm} + 8T_{syed}$
Execution Time	$96.25\ ms$	$23.988\ ms$	$17.831\ ms$	$8.9408\ ms$

Acknowledgements

Authors would like to thank the anonymous reviewers and the editor in chief for their valuable suggestions to improve the quality, correctness, presentation and readability of the manuscript.

References

[1] **S. A. Chaudhry, K. Mahmood, H. Naqvi, M. K. Khan.** An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. *Journal of Medical Systems*, 2015, Vol. 39, No. 11, 1-12.

[2] **S. A. Chaudhry, M. S. Farash, H. Naqvi, S. Kumari, M. K. Khan.** An enhanced privacy preserving remote user authentication scheme with provable security. *Security and Communication Networks*, DOI: 10.1002/sec.1299, 2015.

[3] **M. S. Farash, M. A. Attari.** An enhanced authenticated key agreement for session initiation protocol. *Information Technology and Control*, 2013, Vol. 42, No. 4, 333-342.

[4] **M. S. Farash, M. A. Attari.** An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. *Information Technology and Control*, 2014, Vol. 43, No. 2, 143-150.

[5] **M. S. Farash, M. A. Attari.** 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, 2014.

[6] **M. S. Farash, M. A. Attari.** Cryptanalysis and improvement of a chaotic map-based key agreement protocol using chebyshev sequence membership testing. *Nonlinear Dynamics*, 2014, Vol. 76, No. 2, 1203-1213.

[7] **M. S. Farash, M. A. Attari.** An efficient client password-based authentication scheme with provable security. *The Journal of Supercomputing*, 2014, Vol. 70, No. 2, 1002-1022.

[8] **A. Irshad, M. Sher, E. Rehman, S. Ashraf Ch, M. Ul Hassan, A. Ghani.** A single round-trip SIP authentication scheme for voice over internet protocol using smart card. *Multimedia Tools and Applications*, 2015, Vol. 74, No. 2, 3967-3984.

[9] **Z. Mehmood, N. Uddin, S. Ashraf Ch, W. Nasar, A. Ghani.** An efficient key agreement with rekeying for secured body sensor networks. In: *2012 Second*

International Conference on Digital Information Processing and Communications (ICDIPC), pp. 164-167.

[10] **A. Irshad, M. Sher, M. S. Faisal, A. Ghani, M. Ul Hassan, S. Ashraf Ch.** A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme. *Security and Communication Networks*, 2014, Vol. 7, No. 8, 1210-1218.

[11] **S. Kumari, S. A. Chaudhry, F. Wu, X. Li, M. S. Farash, M. Khurram Khan.** An improved smart card based authentication scheme for session initiation protocol. *Peer-to-Peer Networking and Application*s, DOI: 10.1007/s12083-015-0409-0, 2015.

[12] **S. A. Chaudhry, K. Mahmood, H. Naqvi, M. Sher.** A secure authentication scheme for session initiation protocol based on elliptic curve cryptography. In: *The 13th IEEE International Conference on Dependable, Autonomic and Secure Computing* (DASC 2015), pp 1-5.

[13] **S. A. Chaudhry, H. Naqvi, M. Sher, M. S. Farash, M. Ul Hassan.** An improved and provably secure privacy preserving authentication protocol for SIP. *Peer-to-Peer Networking and Applications*, DOI: 10.1007/s12083-015-0400-9, 2015.

[14] **S. A. Chaudhry, H. Naqvi, T. Shon, M. Sher, M. S. Farash.** Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. Journal of Medical Systems, 2015, Vol. 39, Issue 6.

[15] **Z. Brakerski, V. Vaikuntanathan**. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, 2014, Vol. 43, No. 2, 831-871.

[16] **A. Bogdanov, F. Mendel, F. Regazzoni, V. Rijmen, E. Tischhauser.** ALE: AES-based lightweight authenticated encryption. In: *20th International Workshop on Fast Software Encryption (FSE, 2013)*.

[17] **P. Zhang, C. Lin, Y. Jiang, Y. Fan, X. Shen.** A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks. *IEEE Transactions on Parallel and Distributed Systems*, 2014, Vol. 25, No. 9, 2211-2221.

[18] **Y. Zheng.** Digital signcryption or how to achieve cost (signature & encryption) << cost (signature)+ cost (encryption). In: Advances in Cryptology (CRYPTO'97),pp. 165-179. Springer, 1997.

[19] **M. Toorani , A. A Beheshti.** An elliptic curve-based signcryption scheme with forward secrecy. arXiv preprint arXiv:1005.1856, 2010.

[20] **Y. Han, X. Yang, Y. Hu.** Signcryption based on elliptic curve and its multi-party schemes. In: *Proceedings of the 3rd international conference on Information security,* ACM 2004, pp. 216-217.

[21] **S. A. Ch, M. Sher, A. Ghani, H. Naqvi, A. Irshad.** An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography. *Multimedia Tools and Applications*, 2014, Vol. 74, No. 5, 1711-1723.

[22] **S. A. Ch, N. Nizamuddin, M. Sher.** Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In: *Information Systems, Technology and Management*, pp. 135-142. Springer, 2012.

[23] **N. Nizamuddin, S. A. Ch, N. Amin.** Signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. In: *High Capacity Optical Networks and Enabling Technologies* (HONET), 2011, pp. 244-247.

[24] **N. Nizamuddin, S. A. Ch, W. Nasar, Q. Javaid**. Efficient signcryption schemes based on hyperelliptic curve cryptosystem. In: *7th International Conference on Emerging Technologies* (ICET), 2011, pp. 1-4

[25] **S. Bala, G. Sharma, A. K. Verma**. An improved forward secure elliptic curve signcryption key management scheme for wireless sensor networks. In: *IT Convergence and Security 2012,* pp. 141-149. Springer, 2013.

[26] **S. A. Chaudhry, M. S. Farash, H. Naqvi, M. Sher.** A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. *Electronic Commerce Research*, DOI: 10.1007/s10660-015-9192-5, 2015.

[27] **F. Bao, R. H. Deng.** A signcryption scheme with signature directly verifiable by public key. In: *Public Key Cryptography*, pp. 55-59. Springer, 1998.

[28] **C. Gamage, J. Leiwo, Y. Zheng**. Encrypted message authentication by firewalls. In: *Public Key Cryptography*, pp. 69-81. Springer, 1999.

[29] **Y. Zheng, H. Imai.** How to construct efficient signcryption schemes on elliptic curves. *Information Processing Letters*, 1998, Vol. 68, No. 5, 227-233.

[30] **R.-J. Hwang, C.-H. Lai, F.-Fu Su.** An efficient signcryption scheme with forward secrecy based on elliptic curve. *Applied Mathematics and Computation*, 2005, Vol. 167, No. 2, 870-881.

[31] **M. Toorani, A. A. Beheshti.** An elliptic curve-based signcryption scheme with forward secrecy. arXiv preprint arXiv:1005.1856, 2010.

[32] **E.-J. Yoon, K.-Y. Yoo.** A secure and efficient convertible authenticated encryption scheme with message linkages using elliptic curve cryptosystem. *Applied Mathematical Sciences*, 2013, Vol. 7, No. 127, 6309-6317.

[33] **J.-H. Yang, Y.-F. Chang, Y.-H. Chen**. An efficient authenticated encryption scheme based on ecc and its application for electronic payment. *Information Technology and Control*, 2013, Vol. 42, No. 4, 315-324.

[34] **M. S. Farash**. 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, 2014.

[35] **M. S. Farash, M. A. Attari**. A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. *The Journal of Supercomputing*, 2014, Vol. 69, No. 1, 395-411.

[36] **C.-T. Li.** Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control*, 2011, Vol. 40, No. 2, 157-162.

[37] **J.-W. Hong, S.-Y. Yoon, D.-I. Park, M.-J. Choi, E.-J. Yoon, K.-Y. Yoo**. A new efficient key agreement scheme for VSAT satellite communications based on elliptic curve cryptosystem. *Information Technology and Control*, 2011, Vol. 40, No. 3, 252-259.

[38] **M. S. Farash, M. A. Attari.** 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, 2014.

[39] **Q. Xie, N. Dong, X. Tan, D. S Wong, G. Wang**. Improvement of a three-party password based key exchange protocol with formal verification. *Informa-*

tion Technology and Control, 2013, Vol. 42, No. 3, 231-237.

[40] **Q. Xie, N. Dong, D. S Wong, B. Hu**. Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol. *International Journal of Communication Systems*, DOI: 10.1002/dac.2858, 2014.

[41] **B. Hu, Q. Xie, Y. Li.** Automatic verification of password-based authentication protocols using smart card. In: *International Conference on Information Technology, Computer Engineering and Management Sciences* (ICM),2011, pp. 34-39.

[42] **V. Cheval, B. Blanchet.** Proving more observational equivalences with ProVerif. In: Principles of Security and Trust, pp. 226-246. Springer, 2013.

[43] **H. Oros, C. Popescu.** A secure and efficient offline electronic payment system for wireless networks. *International Journal of Computers, Communications & Control*, 2010, Vol. 5, No. 4, 551-557.

[44] **H. Wang, J. Cao, Y. Zhang.** A flexible payment scheme and its role-based access control. *IEEE Transactions on Knowledge and Data Engineering*, 2005, Vol. 17, No. 3, 425-436.

[45] **H. H. Kilinc, T. Yanik**. A survey of SIP authentication and key agreement schemes. *IEEE Communications Surveys & Tutorials*, 2014, Vol. 16, No. 2, 1005-1023.

Received January 2015.