Secure and Efficient Mobile Payment Using QR Code in an Environment with Dishonest Authority

Xiaoling Zhu(✉), Zhengfeng Hou, Donghui Hu, and Jing Zhang

School of Computer and Information,
Hefei University of Technology, Hefei 230009, China
{zhuxl, houzf, hudh, zhangjing}@hfut.edu.cn

Abstract. Quick response (QR) code payment has become the mainstream of mobile payment in China. However, severe security threat greatly influences consumer confidence. Unifying security and convenience of QR code is a difficult issue. The paper proposes a secure and efficient mobile payment (SEMP) solution where signed and encrypted payment data are embedded into QR code. Since private keys are issued by fully distributed private key generators (PKGs), no matter malicious user, dishonest third party payment platform (TPP), or dishonest PKG, can not impersonate a legal person to authorize a payment or eavesdrop on the communication to obtain privacy information. The scheme has confidentiality and unforgeability. Especially, it can resist against authority attacks. Since no public key certificate is required, it has clear advantage over existing PKI schemes. The comparisons with related schemes show our SEMP scheme maintains less communication cost, while it provides higher security level. So it can better meet security and convenient requirements of mobile payment and it can apply in the QR code payment environment with dishonest authority.

Keywords: Mobile payment · QR code · Security · Signcryption · Authority attacks

1 Introduction

As smartphone is becoming prevalent, mobile payment steps into daily life and brings more convenient services to people. Since quick response (QR) code has the characters of convenient generation, easy publication and quick reading [1], it has been used in payment, advertisement, access control, etc. In China, QR code payment is vigorously promoted by WeChat [2] and Alipay [3], and it has become a mainstream way of mobile payment.

However, when an illegal person embeds malicious URLs into QR code, an ordinary user lacks the ability of detecting malicious URLs, Trojan and virus. If he continues to visit the websites, malicious software will be downloaded and installed quietly. What's worse, if his smartphone infects some payment virus, his money account will suffer serious threats. By modifying color of specific blocks of QR code,

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 452–465, 2016.
DOI: 10.1007/978-3-319-49148-6_37

literature [4] launched a tampering attack. Data leak is also a hidden danger when QR code is plaintext.

Figure 1 shows the application scenario of QR payment. Entities in a QR payment system may include smart phones, payment terminals and third pay platform (TPP). Private key generator (PKG) may be included due to the use of cryptographic techniques. The communication between users and terminals uses QR code and other communications use wireless or wired network. is shown in. When a user scans a QR code from a payment terminal, he has no idea whether the code really comes from a legitimate store; he wishes that the content of the code is not known by attackers. When TPP receives a request for payment, he needs to decide whether it is an authorize payment. Furthermore, a user and a shop both wish that PKG does not leak their secrets. How do they prevent PKG from leaking? In order to solve the upper issues, we propose a secure and efficient mobile payment (SEMP) scheme using QR code. The scheme can ensure confidentiality and integrity of payment data, authentication of payer identity, convenience and non-repudiation of payment operation. Especially, it can resist authority attacks.

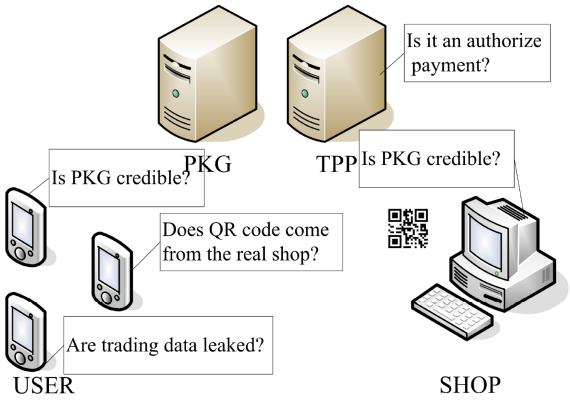

Fig. 1. A scenario of SEMP service

The remainder of this paper is organized as follows. Firstly, we introduce related works in Sect. 2, which will emphasize the motivation of our work. We then describe the framework of our solution in Sect. 3 and the proposed SEMP scheme in details in Sect. 4. We analyze security and performance of our scheme in Sects. 5 and 6, respectively. Finally, we conclude the paper in Sect. 7.

2 Related Works

There exist two patterns in QR application. Active scanning means that a user scans QR code, which is generated by a shop. Otherwise, it is called passive scanning.

For active scanning, checking QR code credibility is an idea. Yao et al. [5] proposed a SafeQR scheme for Andriod phone using Google Safe Browsing API and

Phishtank API; the system should frequently update phishing website list. Literature [6] introduced a third party to detect URL and the burden of the third party is heavy. Milburn et al. [16] presented an identity authentication system, which is based on the SQRL (secure quick reliable login) system by Steve Gibson [17]. In SQRL system, website address and master key are hashed together to create a private key for identity authentication; there are no usernames, passwords or keyboard interaction; if the master key is exposed, the identity unlock key can cancel the master key; however, after analysis, we find if someone steals the smartphone of some person, anyone in possession of the phone can impersonate the person; so complete withdrawal is difficult [16, 17]. Secure design of QR code is another research idea. Czuszynski et al. [7] proposed that the hospital check center encrypts patient data. Lee et al. [8] suggested that a user signs sensitive data with his private key; the communication between a shop and a user requires a payment gateway, which is the bottleneck of system performance. In [8], embedding a public certificate into QR code is a hidden issue due to limited capacity of QR code. How to unify convenience and security deserves attention. For passive scanning, it decreases phishing attacks. But it has the following security threats: (i) money for payment is decided by a shop, and it lacks user's confirmation. (ii) A malicious shop might forge QR code by violent searching for a collision.

Therefore, anti-forgery, anti-leak and convenience are the most concerned issues in mobile payment. In general PKI schemes, the user require obtaining public key certificate before encryption. In identity based encryption (IBE), the public key certificate is not required. IBE has clear advantage over PKI schemes in communication costs [9]. However, PKG in IBE generates all private keys, and he may leak all secrets if he is dishonest, which violates high security requirements of a payment. Some IBE schemes are accountable [10, 11], which eliminate key escrow by combining users and PKG to generate a private key. However, they cannot thoroughly prevent a dishonest PKG from impersonating users.

To solve the above issues, we propose a secure and efficient mobile payment scheme. The main contributions are: (i) Ensure confidentiality and integrity of payment data, authentication of payer identity, non-repudiation of payment operation; (ii) QR code is used to ensure convenience of mobile payment; (iii) The authority can neither forge a signature, nor leak private keys. Our scheme is security and effective when facing a dishonest authority.

3 Overview of the SEMP Scheme

The SEMP scheme supports QR code secure payment when users buy goods from shops. This section provides the system framework, the payment process and security requirements.

3.1 QR Code

QR code is two-dimensional bar code. It can store up to 4296 alphanumeric characters. In our scheme, QR code contains the signed and encrypted payment message.

The message is date || order id || shop id || goods description || total fee. A shop signs and encrypts it, then embeds it into QR code; a user scans and decrypts it. The communication between a user and a shop adopts near field communication; other communications use wireless or wired network. It means that TPP and PKG neither read QR nor generate QR.

3.2 Payment Process

The payment system consists of a user (U), a shop (S), third party payment platform (TPP) and private key generator (PKG).

- A user: A user is equipped with a smartphone, a PDA or a Laptop. He can access the Internet. He can generate and read QR code. His identity, public key and private key are denoted as ID_U, Q_U and D_U, respectively. His phone number or email address may be as ID_U.
- A shop: A shop is equipped with PC terminals. By terminals, he has ability of generating and reading QR code. The identity, public key and private key of the shop are denoted as ID_S, Q_S and D_S, respectively.
- TPP: TPP is an independent agency to protect the interests of both trading parties. If money accounts of trading parties are both on TPP, money is transferred directly from buyer account to seller account. Otherwise, TPP forwards message to a bank. The identity, public key and private key of TPP are denoted as ID_{TPP}, Q_{TPP} and D_{TPP}, respectively.
- PKG: Users, shops and TPP need to obtain their private keys from PKG. If only one authority acts as PKG, abuse occurs. We extend one authority to n authorities forming PKG group $\{P_1, P_2, \ldots, P_n\}$. The extension will increase some costs. The costs generally occur during system setup phase and user registration phase. Since setup occurs once and registration generally occurs once for users, the impact of increased costs is limited.

A user finishes purchasing goods and he comes to a counter for payment. If money accounts of buyers and sellers are on the same TPP, a payment process is as follows.

1. A shop signs and encrypts payment data, embeds them into QR code and shows QR code to a user.
2. The user scans the QR code, extracts data from the QR code, decrypts and verifies the data. If passed, he sends a signed and encrypted payment request to TPP.
3. TPP decrypts and verifies the message. If verification is passed, he transfers money from buyer account to seller account.
4. TPP notices the user money is paid and notices the shop money is received.

When money accounts of buyers and sellers are both on the banks, TPP leads the user to the interface with the bank. The communication between a user and a bank can adopt our proposed signcryption method.

3.3 Security Requirements

We assume that the user, the shop and PKG are all dishonest. They might eavesdrop on the communication to obtain trading information and launch a passive attack. They might forge a signature to obtain illegal money and launch an active attack. So the security requirements are as follows.

Definition 1 (Confidentiality). For attackers, it is computationally infeasible to obtain plaintext from ciphertext.

Definition 2 (Unforgeability). For attackers, it is computationally infeasible to forge a legitimate signature.

Definition 3 (Resistance against authority attacks). For a dishonest authority, it is computationally infeasible to impersonate other entity or leak secrets of others.

Definition 4 (IND-CCA2). A signcryption scheme is semantically secure against chosen ciphertext attack if no probabilistic polynomial time adversary has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the setup algorithm and sends system public parameters to the adversary $\mathcal{A}$.
2. In the first phase, $\mathcal{A}$ makes polynomial bounded number of queries to the following oracles.

   Extract Oracle: $\mathcal{A}$ produces an identity ID_i and queries for the private key. The challenger $\mathcal{C}$ returns the key.
   Signcrypt Oracle: $\mathcal{A}$ produces a message m, a sender identity ID_i and a receiver identity ID_j. $\mathcal{C}$ returns the signcrypted ciphertext.
   Unsigncrypt Oracle: $\mathcal{A}$ produces a sender identity ID_i, receiver identity ID_j, and a signcryption result. $\mathcal{C}$ returns the decrypted result.

3. $\mathcal{A}$ produces two messages m_0 and m_1 of equal length and an arbitrary sender identity ID_A. $\mathcal{C}$ randomly chooses a bit $u \in \{0,1\}$ and computes the signcryption σ^*, and returns σ^* to $\mathcal{A}$ as a challenge.
4. $\mathcal{A}$ is allowed to make polynomial bounded number of new queries as in Step 2 with the restrictions that it should not query Unsigncryption Oracle for σ^* and Extract Oracle for ID_B.
5. At the end of this game, outputs a bit u', $\mathcal{A}$ wins the game if $u = u'$.

4 Description of the SEMP Scheme

Since smartphone is a resource constrained device, it cannot bear much burden. Signcryption can complete digital signature and public key encryption at the same time, and its communication and computation costs might be lower. In the IBE, the public key is directly from the identity, and the user does not need to get public key certificate. So based on IBE, we design a signcryption scheme. It is derived from the IBE proposed

by Boneh and Franklin [12] and Paterson [13] and a distributed structure proposed by Feldman [14]. In the scheme users need one time scan to complete the communication with payment terminal.

Definition 5 (Bilinear map). Let G_1 be a cyclic additive group with a generator P, whose order is a prime q, and G_2 be a cyclic multiplicative group with the same order q. A map $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map if following properties are satisfied: bilinearity, non-degeneracy and computability.

4.1 Setup

Let $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^l \times \{0,1\}^* \rightarrow Z_q^*$, $H_3 : G_1 \rightarrow Z_q^*$ and $H_4 : G_2 \rightarrow \{0,1\}^l$ be collision-resistant functions.

1. Each member P_i $(1 \leq i \leq n)$ in PKGs randomly picks a secret $d_i \in Z_q^*$ as his member private key, computes and broadcasts his member public key $P_{pub_i} = d_i P$ and further computes the group public key $P_{pub} = \sum_{i=1}^{n} P_{pub_i}$. He chooses a random polynomial $f_i(x) = f_{i,0} + f_{i,1}x + f_{i,2}x^2 + \ldots + f_{i,t-1}x^{t-1}$ over Z_q^*, where $f_i(0) = d_i$.
2. P_i computes $s_{i,j} = f_i(ID_j) \bmod q (1 \leq j \leq n)$ and sends $s_{i,j}$ to P_j secretly. He broadcasts $F_{i,l} = f_{i,l}P (1 \leq l \leq t-1)$.
3. P_j receives $s_{i,j}$ from other members and verifies $s_{i,j}P = \sum_{l=0}^{t-1} F_{i,l} \cdot ID_j^l$ If the validation passes, he computes the share $s_j = \sum_{i=1}^{n} s_{i,j} \bmod q$ and broadcasts s_jP. s_jP will be accepted by other members if $s_jP = \sum_{l=0}^{t-1} F_l ID_j^l$, where $F_l = \sum_{i=1}^{n} F_{i,l}$. Otherwise, P_j will be complained. When the complaint number achieves a threshold value, P_j will be added to a blacklist.

After implementing the steps, each member obtains public parameters $\{P, P_{pub}, \{s_iP\}_{1 \leq i \leq n}\}$ and his secrets $\{s_i, d_i\}$. The group private key $d = \sum_{i=1}^{n} d_i$ is owned jointly by PKGs; any single member does not know d. The group public key P_{pub} satisfies $P_{pub} = dP$ since $P_{pub} = \sum_{i=1}^{n} P_{pub_i} = \sum_{i=1}^{n} d_i P = dP$. The two verification equations in Step 3 are clearly established since $s_{i,j}P = f_i(ID_j)P = f_{i,0}P + f_{i,1}P \cdot ID_j + \ldots + f_{i,t-1}P \cdot ID_j^{t-1} = \sum_{l=0}^{t-1} F_{i,l} \cdot ID_j^l$ and $s_jP = \sum_{i=1}^{n} s_{i,j}P = \sum_{i=1}^{n}\sum_{l=0}^{t-1} F_{i,l} \cdot ID_j^l = \sum_{l=0}^{t-1}\sum_{i=1}^{n} F_{i,l} \cdot ID_j^l = \sum_{l=0}^{t-1} F_l \cdot ID_j^l$.

The protocol is implemented among PKGs. It increases the burden of PKG. Since the setup protocol occurs once, the impact of the increased costs is limited.

4.2 Registration

A user, a shop or TPP needs to obtain his private key from t participating members of PKGs, namely, P_1, P_2,…, P_t. First, the applicant sends his identification ID to t participating members.

1. Each participating member P_i $(1 \leq i \leq t)$ computes $s_i Q_{ID}$ and sends it to the applicant secretly. Here, $Q_{ID} = H_1(ID)$.
2. Assume the applican has obtained member public key $s_i P$, and he verifies $e(s_i Q_{ID}, P) = e(Q_{ID}, s_i P)$. If passed, he computess the private key $D_{ID} = \sum_{i=1}^{t} l_i(0) s_i Q_{ID}$, where $l_i(0) = \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{j}{j-i}$ and $D_{ID} = \sum_{i=1}^{t} (\sum_{j=1}^{n} s_{j,i}) l_i(0) Q_{ID} = (\sum_{j=1}^{n} (\sum_{i=1}^{t} s_{j,i} l_i(0))) Q_{ID} = (\sum_{j=1}^{n} d_j) Q_{ID} = dQ_{ID}$.

After interaction between the applicant and t participating members of PKGs, the applicant obtains his private key D_{ID}, i.e., the user obtains D_U, the shop obtains D_S or TPP obtains D_{TPP}. In the above process, even if m participating members $(m < t)$ launch collusion attack, they cannot obtain a legitimate D_{ID}. Our registration protocol maintains good property of security in the presence of dishonest authorities.

4.3 Bill Generation

When a user comes to a counter for payment, a shop generates a payment list and signs it. The payment list is denoted by m, where m = date ‖ order id ‖ shop id ‖ goods description ‖ total fee. Their byte length is 2, 16, 10, 100 and 8, respectively.

1. A shop picks a random number $r \in Z_q^*$, computes $R = rP$ and makes a signature

$$S = \frac{H_2(m, ID_S)P + H_3(R)D_S}{r + H_2(m, ID_S)} \tag{1}$$

where ID_S and D_S are the identity and the private key of the shop, respectively.

2. The shop computes $w = e(Q_U, P_{pub})^r$, where $Q_U = H_1(ID_U)$, ID_U and Q_U is the identity and the public key of the user respectively. Then, the shop encrypts m and obtains the cipher

$$c = H_4(w) \oplus m \tag{2}$$

He further embeds the results (c, R, S) into QR code and shows QR code to the user.

4.4 Bill Payment

1. The user scans QR code and obtain (c, R, S).
2. He computes $w = e(D_U, R)$, where D_U is his private key. Then he makes a decryption $m = H_4(w) \oplus c$.
3. The user verifies

$$e(S, R + H_2(m, ID_S)P) = e(P, P)^{H_2(m, ID_S)} e(Q_S, P_{pub})^{H_3(R)} \tag{3}$$

where $Q_S = H_1(ID_S)$.

4. If verification passes, the user generates the payment request m' = date ‖ order id ‖ shop id ‖ user id ‖ total fee.
5. He picks a random number $r' \in Z_q^*$, computes $R' = r'P$ and makes a signature $S' = \frac{H_2(m', ID_U)P + H_3(R')D_U}{r + H_2(m', ID_U)}$. He computes $w' = e(Q_{TTP}, P_{pub})^r$ and $c' = H_4(w') \oplus m'$ and submits a payment request (c', R', S') to TPP.
6. TPP receives (c', R', S'), computes $w' = e(D_{TTP}, R')$ and $m' = H_4(w') \oplus c'$, further verifies $e(S', R' + H_2(m', ID_U)P) = e(P, P)^{H_2(m', ID_U)} e(Q_U, P_{pub})^{H_3(R')}$, where D_{TPP} is his private key. If passed, TPP transfer money from the user account to the shop account.

Equation (3) is correct since $e(S, R + H_2(m, ID_S)P) = e(\frac{H_2(m, ID_S)P + H_3(R)D_S}{r + H_2(m, ID_S)}, (r + H_2(m, ID_S))P) = e(H_2(m, ID_S)P, P)\ e(H_3(R)D_S, P) = e(P, P)^{H_2(m, ID_S)} e(Q_S, P_{pub})^{H_3(R)}$.

5 Security Analysis

Definition 6 (Computational bilinear Diffie-Hellman (CBDH) problem). Given $P \in G_1$, aP, bP, cP for some unknowns $a, b, c \in Z_p^*$, find $e(P, P)^{abc}$.

Definition 7 (CBDH assumptions) The advantage of any probabilistic polynomial time algorithm in solving the CBDH problem is negligibly small, i.e., CBDH problem is assumed to be hard.

Proposition 1. *Our scheme is secure against any IND-CCA2 adversary under the random oracle model and CBDH assumption.*

Proof. The challenger $\mathcal{C}$ receives an instance (P, aP, bP, cP) of the CBDH problem. His goal is to compute $e(P, P)^{abc}$. We expect that $\mathcal{C}$ can use an IND-CCA2 adversary $\mathcal{A}$ to solve the CBDH problem. $\mathcal{C}$ gives $\mathcal{A}$ public parameters $\{P, P_{pub} = cP\}$. The descriptions of some oracles are as follows.

- $H_1(ID_i)$: $\mathcal{C}$ checks whether there is a tuple (ID_i, Q_i) in list L_1. If it exists, $\mathcal{C}$ returns Q_i to $\mathcal{A}$. Otherwise, $\mathcal{C}$ does the following: If $ID_i = ID_B$, $\mathcal{C}$ returns $Q_i = bP$; else chooses a random number $x \in Z_q^*$ and returns $Q_i = xP$. Then, add (ID_i, Q_i) to L_1.
- $H_2(m, ID_i)$: $\mathcal{C}$ checks whether there is a tuple (m, ID_i, h_2) in L_2. If it exists, $\mathcal{C}$ returns h_2. Otherwise, C chooses a random number h_2, adds (m, ID_i, h_2) to L_2 and returns h_2.

- $H_3(R)$: $\mathcal{C}$ checks whether there is a tuple (R, h_3) in L_3. If it exists, $\mathcal{C}$ returns h_3. Otherwise, $\mathcal{C}$ chooses a random number h_3, adds (R, h_3) to L_3 and returns h_3.
- $H_4(w)$: $\mathcal{C}$ checks whether there is (w, h_4) in L_4. If it exists, $\mathcal{C}$ returns h_4. Otherwise, $\mathcal{C}$ chooses randomly a l -bit integer h_4, adds (R, h_4) to L_4 and returns h_4.
- Extract (ID_i): If $ID_i = ID_B$, return stop simulation. Otherwise, get (ID_i, Q_i) through H_1 and return $D_i = cQ_i$.
- Signcrypt (m, ID_i, ID_j):
  - $ID_i \neq ID_B$. Get the private key D_i by running Extract Oracle. Choose a random number r. Compute $R = rP$. Get a tuple (m, ID_i, h_2) through H_2 and (R, h_3) through H_3. Compute $S = \frac{h_2 P + h_3 D_i}{r + h_2}$. Get (ID_j, Q_j) through H_1. Compute $w = e(Q_j, P_{pub})^r$. Get (w, h_4) through H_4. Compute $c = h_4 \oplus m$. Finally, return signcryption results (c, R, S).
  - $ID_i = ID_B$. Choose random numbers r and k. Compute $R = rP$ and $S = kP$. Get (ID_j, Q_j) through H_1. Compute $w = e(Q_j, P_{pub})^r$. Get (w, h_4) through H_4. Compute $c = h_4 \oplus m$. Return signcryption results (c, R, S).
- Unsigncrypt(c, R, S, ID_i, ID_j):
  - $ID_j \neq ID_B$. Get D_j through Extraction oracle. Compute $w = e(D_j, R)$. Get (w, h_4) through H_4. Compute $m = h_4 \oplus c$. Get (ID_i, Q_i) through H_1, (m, ID_i, h_2) through H_2 and (R, h_3) through H_3. Verify $e(S, R + h_2 P) = e(P, P)^{h_2} e(Q_i, P_{pub})^{h_3}$. If verification does not pass, $\mathcal{C}$ stops simulation. Otherwise, $\mathcal{C}$ returns m.
  - $ID_j = ID_B$. Traverse each tuple (w, h_4) in L_4 and compute $m = h_4 \oplus c$. Get (ID_i, Q_i) through H_1, (m, ID_i, h_2) through H_2 and (R, h_3) through H_3. Verify $e(S, R + h_2 P) = e(P, P)^{h_2} e(Q_i, P_{pub})^{h_3}$. If the above equation holds for a certain tuple, then $\mathcal{C}$ returns related m. If not passed for all tuples in L_4, $\mathcal{C}$ stops simulation.

After the first stage, $\mathcal{A}$ outputs two plaintexts m_0 and m_1, $\mathcal{C}$ chooses $u \in \{0,1\}$ and signcrypts m_u. Assume $R^* = aP$ and $w = h$, then $\mathcal{C}$ return $\sigma^* = (c^*, R^*, S^*)$ to $\mathcal{A}$. $\mathcal{A}$ performs a second series of queries which is treated in the same way as the first one. At the end of the simulation, $\mathcal{A}$ returns a bit u' to $\mathcal{C}$ for which he believes the relation σ^* = Signcrypt $(m_u' ID_i, ID_j)$ holds. If $u = u'$, $\mathcal{C}$ outputs $h = e(R^*, D_B) = e(aP, cbP) = e(P, P)^{abc}$ as a solution of the CBDH problem, otherwise $\mathcal{C}$ stops.

If there is an adversary who can succeed in such a CCA2 attack with non-negligible advantage, that means there is an algorithm to solve the CBDH problem with non-negligible advantage. The scheme is secure against any IND-CCA2 attack under CBDH assumption.

Proposition 2. *Our scheme has the existential unforgeability against adaptive chosen messages attacks under the random oracle model and CBDH assumption.*

Proof. The scheme is based on Paterson's scheme [13] and Paterson's scheme can resist existential forgery against adaptive chosen messages attacks.

Proposition 3. *Our scheme can resist authority attacks under CBDH assumption.*

Proof. Since the hardness of the discrete logarithm problem, s_iQ_{ID} cannot leak out s_i and P_{pub} cannot leak out d. For a given identity ID, at least t members are needed to issue a valid private key. Therefore, any entity, even including PKG and TPP, cannot impersonate others to forgery valid signature.

We compare our scheme with similar works that are intended to ensure security of

Table 1. Secutity features comparisons

	Confidentiality	Resist forgery	Resist authority attacks
Czuszynski et al.'s scheme [7]	Yes	No	No
Lee et al.'s scheme [8]	No	Yes	No
Milburn et al.'s scheme [16]	Yes	Yes	No
Our scheme	Yes	Yes	Yes

QR code. The results of comparisons of security features are shown in Table 1. Czuszynski et al. [7] used AES algorithm to encrypt data for confidentiality. Lee et al. [8] made a digital signature on a payment list, which is unforgeability. However, a trust center exists in the two schemes: the check center for encryption in [7] and CA in [8]. The schemes are suffered attacks from a trust center. Milburn et al. [16] used AES encryption and ECDSA signature, which achieves confidentiality and unforgeability. In [16], a user issues private key and public key by himself and no third party knows the private key. It seems secure. But a server can also issue private key and public key, and then claims that the keys are issued by the user. For any assessment institution, he cannot distinguish the keys issued by the user from the keys issued by the server. So the method is still unable to resist this kind of authority attacks. In our scheme, the payment list is signed and encrypted. Private keys are generated by distributed PKGs. The collusion of less than n members cannot know the private key and further forge a valid signature. Therefore, our scheme has confidentiality, unforgeability and resistance against authority attacks.

6 Performance Analysis

For convenience to evaluate the computation costs of the scheme, we ignore some operations such as a hash function and a multiplication operation since they are quite lighter in terms of load. We focused on some time-consuming operations defined in the following notations. T_P denotes the time of executing a bilinear map operation. All exponentiations in G_2 can be transformed into scalar multiplications in G_1 to get a fast implementation of a bilinear map. So we use T_{G1} to represent the time of executing a scalar multiplication or an exponentiation operation. To evaluate the communication costs, $|q|$, $|G_1|$, $|c|$ and $|c'|$ denote the length of the order of G_1, the element in G_1, the cipher c and c', respectively.

6.1 Performance of the SEMP Scheme

Table 2 shows computation and communication costs of SEMP scheme during four different phases, i.e., setup, registration, bill generation and bill payment. In the table, n is the number of total members in PKGs and t is the number of members participating to issue private keys.

Table 2. Computation and communication costs of the SEMP scheme

	Setup	Registration	Bill generation	Bill payment
Computation costs	$((n–1)$ $(2t + 1) + t + 1)T_{G1}$	$2tT_P + 2tT_{G1}$	$T_P + 4T_{G1}$	$9T_P + 11T_{G1}$
Communication costs	$(t + 1)\|G_1\| + (n–1)\|q\|$	$2t\|G_1\|$	$\|c\| + 2\|G_1\|$	$\|c'\| + 2\|G_1\|$

6.2 Performance Comparisons with Our Schemes

To achieve the similar security level of 1024 bits RSA signature, Literature [15] proposed $|q|$ = 160 bits = 20 bytes and $|G_1|$ = 161bits ≈ 20 bytes; it requires 4.5 ms to perform a bilinear map and 0.6 ms to perform a scalar multiplication in G_1. For elliptic curve digital signature algorithm (ECDSA), if the key is 28 bytes, then ECDSA signature is 53 bytes; the point on the elliptic curve is 29 bytes; public certificate is 84 bytes. It requires 0.8 ms to perform a signature and 4.2 ms to perform a verification [18]. For AES algorithm, it requires 94 μs to perform encryption, the same with decryption [19]. Raya proposed that when HMAC is based on SHA-224, the output is 28 bytes and the operation time is 28 μs [18]. From the definition of the bill list in Sect. 4.3 and the payment list in Sect. 4.4, we obtain $|m|$ = 2 + 10 + 16 + 200 + 8 = 236 (bytes) and $|m'|$ = 2 + 16 + 10 + 10 + 8 = 46 (bytes). Since $c = H_4(w) \oplus m$, $|c|$ = 236 bytes. Similarly, $|c'|$ = 46 bytes.

In the following, we shall mainly compare our SEMP scheme with Lee et al.'s scheme [8] and Milburn et al.'s scheme [16]. The framework of Lee et al.'s scheme is similar to ours. Though Milburn et al.'s scheme is mainly used in identify authentication environment and the system framework is not similar to ours, we extend it to a QR payment environment.

In Lee et al.'s scheme [8], both a user and a shop have to obtain their public certificates from CA during the initialization. Then they register themselves to payment gateway (PG) using certificate. When finishing shopping, payment information and digital signature are transmitted to PG. After verification, the shop shows shop number, payment number and digital signature value in the form of QR to users. Then the user downloads payment information from PG. During payment, he signs payment data and transmits them to PG. In Milburn et al.'s scheme [16], the SQRL app hashes the website address and master key together to create a private key. The identity of the user is proved by the digital signature with the private key. There is no third party

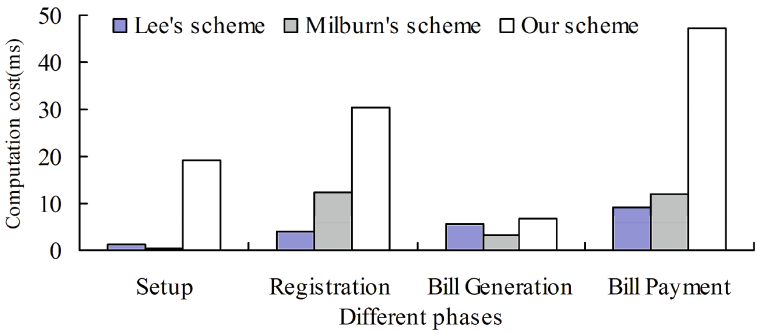

Fig. 2. The computation cost comparisons in different phases

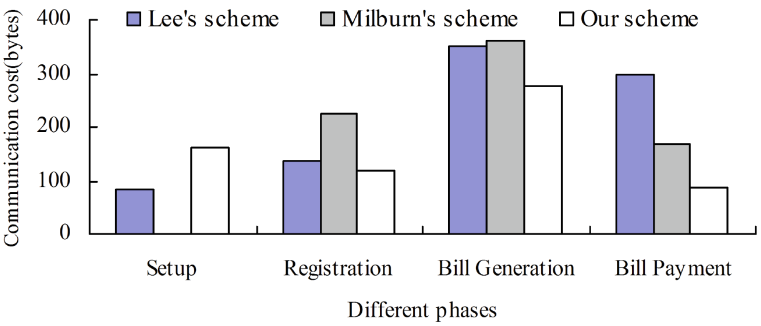

Fig. 3. The communication cost comparisons in different phases

involvement in the authentication process. When master key leaks, SQRL identity lock uses Diffie-Hellman key agreement to revoke it.

Figures 2 and 3 show communication and computation cost comparisons during different phases, respectively. Here, $n = 5$ and $t = 3$ in our scheme. We observe that setup phase in our scheme requires more computation and communication costs, compared with the existing solutions [8, 16]. It is because we adopt a distributed key generator structure to resist authority attacks and a detection method to find dishonest authority nodes. Considering the setup protocol generally performs one time in the system, its influence is limited. In addition, bill payment phase in our scheme requires more computation cost. It is because the phase requires 9 bilinear map operations, which is time-consuming. For communication cost, our SEMP scheme has better performance to the existing solutions [8, 16] during registration, bill generation and bill payment, while providing higher security level, especially in the aspect of resisting authority attacks.

7 Conclusion

Anti-forgery, anti-leak and convenience are the most concerned issues in mobile payment. In this paper, we formalized the definition and secure payment model. Subsequently, we proposed a SEMP scheme. In the scheme, payment data are signed and encrypted based on IBE and private keys are issued by fully distributed PKGs. Malicious users, dishonest TPP or dishonest PKG cannot impersonate a legal user to authorize a payment. Our scheme has confidentiality, unforgeability and resistance against authority attacks. Since no public key certificate is required, it has clear communication advantage over PKI schemes. Security analysis and performance analysis show that it has high security and convenience and it can be applied in mobile payment efficiently.

For future research, we will discuss how to put the scheme into a practical system to satisfy specific security and application requirements of mobile payment.

Acknowledgments. This work was supported by the Natural Science Foundation of Anhui Province (Grant No.1608085MF141), by the Fundamental Research Funds for the Central Universities (Grant No. J2014HGBZ0131) and by the Humanity and Social Science Key Foundation of Anhui Province (Grant No. SK2015A578).

References

1. Krombholz, K., Frühwirt, P., Kieseberg, P., Kapsalis, I., Huber, M., Weippl, E.: QR code security: a survey of attacks and challenges for usable security. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2014. LNCS, vol. 8533, pp. 79–90. Springer, Heidelberg (2014). doi:10.1007/978-3-319-07620-1_8
2. Tencent Inc. (2016). https://wx.qq.com/
3. Alibaba Group (2016). https://www.alipay.com/
4. Shah, D., Shah, Y.: QR code and its security issues. Int. J. Comput. Sci. **2**(11), 22–26 (2014)
5. Yao, H., Shin, D.: Towards preventing qr code based attacks on android phone using security warnings. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, pp. 341–346 (2013)
6. Wang, C.D., Feng, C.R., Gao, S.M.: Research on the security of two-dimension code used in the mobile payment. J. Tian Jin Univ. Technol. **30**(3), 15–20 (2014)
7. Czuszynski, K., Ruminski, J.: Interaction with medical data using QR-codes. In: Human System Interactions (HSI), pp. 182–187 (2014)
8. Lee, J., Cho, C.H., Jun, M.S.: Secure quick response-payment (QR-Pay) system using mobile device. In: Advanced Communication Technology (ICACT), pp. 1424–1427 (2011)
9. Han, J., Yang, Y., Huang, X., Yuen, T.H., Li, J., Cao, J.: Accountable mobile E-commerce scheme via identity-based plaintext-checkable encryption. Inf. Sci. **345**, 143–155 (2016)
10. Goyal, V.: Reducing trust in the PKG in identity based cryptosystems. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 430–447. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74143-5_24
11. Libert, B., Vergnaud, D.: Towards practical black-box accountable authority IBE: weak black-box traceability with short ciphertexts and private keys. IEEE Trans. Inf. Theor. **57**(10), 7189–7204 (2011)

12. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_13
13. Paterson, K.G.: ID-based signatures from pairings on elliptic curve. Electron. Lett. **38**(18), 1025–1026 (2002)
14. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: Foundations of Computer Science, pp. 427–438 (1987)
15. Chen, L., Ng, S.L., Wang, G.: Threshold anonymous announcement in VANETs. Sel. Areas Commun. **29**(3), 605–615 (2011)
16. Milburn, J., Lee, H.: FassKey: a secure and convenient authentication system. In: IEEE Netsoft Conference and Workshops, pp. 489–495 (2016)
17. Steve, G.: SQRL–Secure Quick Reliable Login (2013). https://www.grc.com/sqrl/sqrl.htm
18. Raya, M., Hubaux, J.P.: Securing vehicular ad hoc networks. J. Comput. Secur. **15**(1), 39–68 (2007)
19. Calandriello, G., Papadimitratos, P., Hubaux, J.P., Lioy, A.: On the performance of secure vehicular communication systems. IEEE Trans. Dependable Secure Comput. **8**(6), 898–912 (2011)