Mutual Heterogeneous Signcryption Schemes for 5G Network Slicings

JINGWEI LIU [1], (Member, IEEE), **LIHUAN ZHANG** [1], **RONG SUN** [1], (Member, IEEE), **XIAOJIANG DU** [2], (Senior Member, IEEE), AND **MOHSEN GUIZANI** [3], (Fellow, IEEE)

[1] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
[2] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[3] Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID 83844, USA

Corresponding author: Jingwei Liu (jwliu@mail.xidian.edu.cn)

ABSTRACT With the emergence of mobile communication technologies, we are entering the fifth generation (5G) mobile communication system era. Various application scenarios will arise in the 5G era to meet the different service requirements. Different 5G network slicings may deploy different public key cryptosystems. The security issues among the heterogeneous systems should be considered. In order to ensure the secure communications between 5G network slicings, in different public cryptosystems, we propose two heterogeneous signcryption schemes which can achieve mutual communications between the public key infrastructure and the certificateless public key cryptography environment. We prove that our schemes have the indistinguishability against adaptive chosen ciphertext attack under the computational Diffie–Hellman problem and the existential unforgeability against adaptive chosen message attack under the discrete logarithm problem in the random oracle model. We also set up two heterogeneous cryptosystems on Raspberry Pi to simulate the interprocess communication between different public key environments. Furthermore, we quantify and analyze the performance of each scheme. Compared with the existing schemes, our schemes achieve greater efficiency and security.

INDEX TERMS 5G network slicing, signcryption, heterogeneous system, chosen ciphertext attack, unforgeability, interprocess communication.

I. INTRODUCTION

Since the emergence of wireless networks, the Mobile Internet has experienced explosive growth four times (1G, 2G, 3G, 4G). It has become the foundation of information networks connecting the human society. The existing traditional communication services are difficult to adapt to many application scenarios. Therefore, the fifth generation mobile communication technology arises at this historic moment. 5G is a hot spot in global research. In 2012, the European Union officially launched METIS (mobile and wireless communications enables for the 2020 information society) project [1]. In Asia, South Korea started the "GIGA Korea" 5G project [2] in 2013. Chinese IMT-2020 promotion group and the "863" plan were also launched in June, 2013 and March, 2014 respectively [3]. Due to 5G services having the advantages of faster speed, larger capacity, and lower cost, many scholars all over the world are carrying out a wide range of research on the candidate frequency bands of

5G communication, the development of 5G vision, 5G's application requirements and key technologies [4], while, at the same time they also raise more security challenges. People who use 5G's services face more extensive and complex security threats.

Since a variety of 5G application scenarios may have different requirements, 5G network resources are divided into different network slicings to meet these demands. Dividing the network can reduce the network operating cost and improve the efficiency.

A. NETWORK SLICING
The 5G system is composed of three layers shown as follows. They correlate to each other through the entities of network layout [5]:
- Infrastructure Resource Layer: Through virtualization principles, the physical resources of the fixed and mobile convergence network are exposed to the
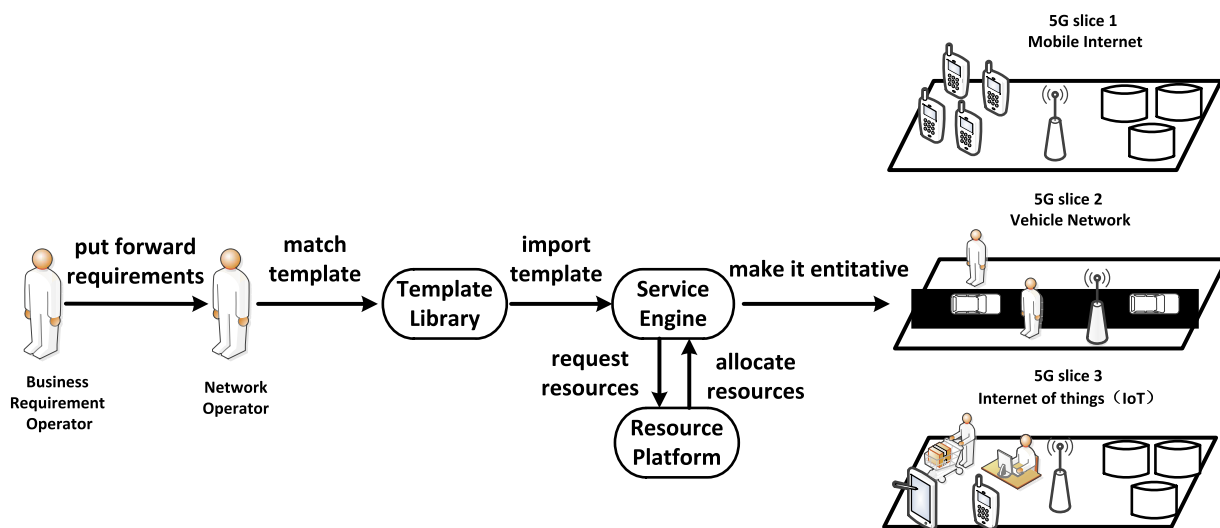
FIGURE 1. The system architecture of 5G.

orchestration entity. These resources are composed of access nodes, cloud nodes, 5G devices, network nodes and related links.

- Business Enablement Layer: All the functions of the convergence network should be constructed in a modular form and documented to the database. The functions of the software module and the configuration parameters of the specific network parts can be downloaded from the resource database.
- Business Application Layer: This layer deploys the specific applications and services in 5G network [6].

In 2015, Ericsson proposed that 5G systems would be built to logical network slicings, which can enable operators to meet the wide range of users' demands. The network slicings are also called the ''5G network slicings''. They comprise a group of network functions, resources, connection relationships and typically covering multiple technical domains including terminals, access networks, *etc*. Through the virtual independent logic network infrastructure, the 5G network slicings technology provides an isolated network environment for different applications. In this way, a wide variety of scenarios can be customized according to the demands of network functions and characteristics. Different network slicings include different proprietary networks with separate logics. A 5G slicing is composed of various functions and specific Radio Access Technology (RAT) sets [7]. It can span all domains of the network: software modules running on the cloud, specific configurations of the transport network supporting flexible and accurate positioning, dedicated radio configurations, etc. The specific application scenarios or business models can decide the utilization forms of network functions and the compound modes of RAT sets.

Via the virtualization technology, network infrastructure resources are virtualized into a number of proprietary networks according to the requirements of specific applications. Slicings can customize network functions and manage

network resources based on different business scenarios [8]. Each network slicing can be abstracted as a logical network formed by the collection of network functions and their corresponding configurations [9]. These logical networks (different 5G slicings) can provide network services accordingly.

There are three steps in a slicing's life cycle: the creation, the management, and the revocation. As shown in Fig. 1, the business requirement operator puts forward requirements to the network operator, upon receiving these requirements, the network operator matches the network slicing template according to the requirements of the business scenario. A slicing template contains the network function components required, the component interaction interfaces, and the description of the network resources. When the template is imported, the service engine can apply for network resources from the resource platform. After acquiring the resources, the service engine can use them to achieve virtual network functions and make it entitative [10]. Fig. 1 shows that 5G network slicings could be the Mobile Internet, Internet of Vehicles, Internet of Things, *etc*.

B. RELATED WORK

In a traditional PKI cryptosystem, there is a Certification Authority (CA) that issues public key certificates for each user and binds certificates with their identities. However, as the number of users increases, this method uses a great deal of time and storage space in certificate management. In order to solve this problem, Identity-Base Cryptosystem (IBC) [11] was proposed by Shamir in 1984. In IBC, users' public keys are their identities and private keys are generated by Key Generation Centers (KGC). However, it leads the issue of key escrow; because of this issue, some key management schemes have been proposed to address this problem [12], [13]. In [14], CLC was also introduced to solve this problem in which the private key is formed by two parts. One part is the secret value of the user's choice and the other is the partial

private key issued by KGC. Hence, KGC has no way to get the full private key, so the key escrow problem is solved effectively.

In order to achieve confidentiality, integrity, authentication and nonrepudiation simultaneously, a traditional approach is first to sign a message and then to encrypt it, called the signature-then-encryption approach. For the sake of optimizing algorithm and improving efficiency, the concept of signcryption was first introduced by Zhang [15] in 1997, and the formal security model of signcryption [16] was first proposed in 2002. Signcryption is a new cryptographic primitive that fulfills the functions of digital signature and public key encryption in a single logical step, at a cost significantly lower than that required by the traditional signature-then-encryption approach. Although some signcryption schemes based on PKI or CLC [17], [18] were proposed in the past few years, they are only suitable for homogeneous environments. Research on heterogeneous communications has been considered by scholars around the world, in order to facilitate communications [19]–[21]. Similarly, in the 5G system, different network slicings may deploy different public cryptosystems. In order to ensure the secure communications between 5G network slicings over different public key systems, we put forward two innovative mutual heterogeneous signcryption schemes.

In 2010, Sun and Li [22] proposed a heterogeneous signcryption scheme between PKI and IBC and discussed it in the multi-receiver setting, however their scheme could only resist outside attacks, and it did not satisfy the non-repudiation. In 2011, Huang *et al.* [23] proposed a heterogeneous signcryption scheme for PKI-to-IBC that can achieve the inside security. Regardless, it only permits a sender in the IBC to transmit a message to a receiver in the PKI and does not provide the mutual communications. In 2013, Li *et al.* [24] proposed two signcryption schemes that provided bidirectional communication between PKI and IBC. This scheme relied on the cumbersome pairing operations which lead to the inefficiency. In 2016, Li *et al.* [25] proposed a heterogeneous signcryption to design an access control scheme without certificates. In 2017, a concrete heterogeneous signcryption scheme of IBC-to-CLC was presented in [26]. In the same year, Wang *et al.* [27] proposed an IBC-to-PKI heterogeneous signcryption scheme in the standard model.

In this paper, we propose two mutual heterogeneous signcryption schemes between PKI and CLC public cryptosystems named PCHS (PKI-CLC Heterogeneous Signcryption) and CPHS (CLC-PKI Heterogeneous Signcryption). When the users in the 5G slicing based on PKI environment (such as a Mobile Internet slicing) try to communicate with the users in the 5G slicing based on CLC (such as a Vehicle Internet slicing), they can use the PCHS scheme to establish a secure communication, if in the opposite case, the CPHS scheme can be used.

The rest of paper is organized as follows. We introduce the preliminary work e.g. the generic signcryption model and the complexity assumptions in Section II. Then, we propose

two efficient mutual heterogeneous signcryption schemes in Section III. The security analysis of the proposed schemes are given in Section IV. The performance is evaluated in Section V. Section VI concludes this paper.

II. PRELIMINARIES
A heterogeneous signcryption scheme generally consists of the following five algorithms:

Setup: This is a probabilistic algorithm running by Private Key Generator (PKG). It inputs a security parameter l, and outputs the system parameters and the master key. PKG publishes the system parameters while keeping the master key s in secret.

PKI-KG: This is a key generation algorithm for PKI users. Each user chooses his/her private key SK_p and publishes the public key PK_p.

CLC-KG: This is a key generation algorithm for CLC users.
- Partial Private Key Extract: The user inputs the system parameters, the master key s and his/her identity ID, PKG outputs the partial private key d and transmits it to the user in a secure way.
- Set Secret Value: The user inputs an identity ID, and outputs a secret value x_c.
- Private Key Extract: The user inputs a partial private key d and a secret value x_c, and outputs a full private key SK_c.
- Public Key Extract: The user inputs an identity ID and the secret value x_c, and outputs a public key PK_c.

Signcrypt: This is a probabilistic signcryption algorithm running by a sender. It takes a message m, the sender's private key SK_p and a receiver's public key PK_c, then outputs the ciphertext σ.

Unsigncrypt: This is a deterministic unsigncryption algorithm running by a receiver. It takes the ciphertext σ, the sender's public key PK_p and the receiver's secret key SK_c. Then, it outputs the plaintext m, or the symbol $\perp$ if σ is an invalid ciphertext between the specific sender and receiver.

A. BILINEAR PAIRINGS
The bilinear pairing namely Weil pairing or Tate pairing of algebraic curves is defined as a map $e : G_1 \times G_1 \rightarrow G_2$. Here, G_1 is a cyclic additive group of a large prime order q. P is a generator of G_1. G_2 is a cyclic multiplicative group of the same order q. Let a and b be elements in Z_q^*. A bilinear pairing has the following properties:
- Bilinearity: Let $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P, Q \in G_1$ and $a, b \in Z_q^*$. This can be related as $\forall P, Q, S \in G_1$, $e(P + Q, S) = e(P, S)e(Q, S)$ and $e(P, Q + S) = e(P, Q)e(P, S)$.
- Nondegeneracy: There exist $P, Q \in G_1$, such that $e(P, Q) \neq 1_{G_2}$. Here, 1_{G_2} denotes the identity element of G_2.
- Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.
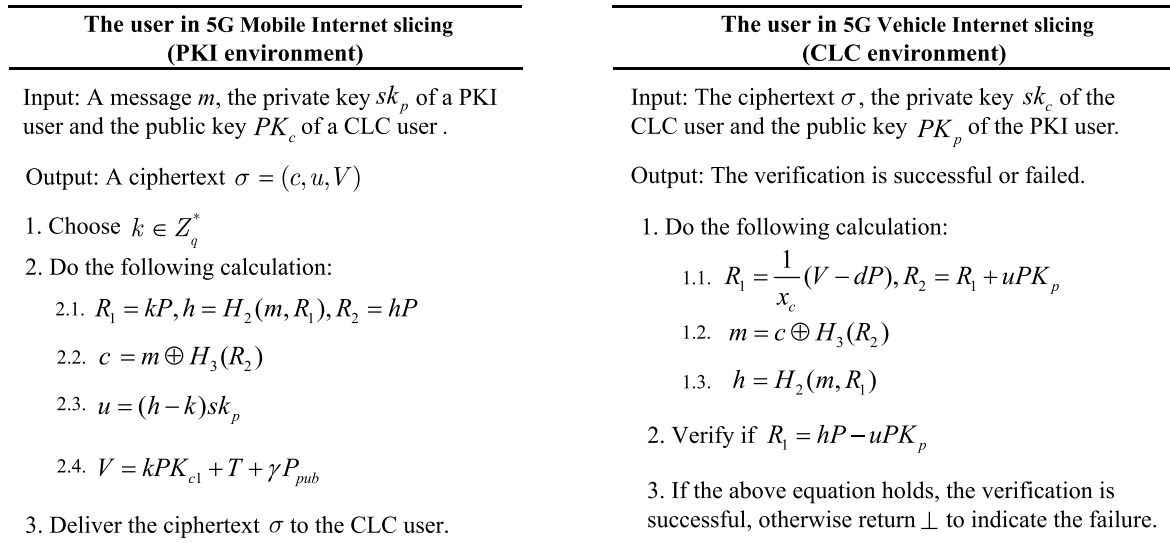
The user in 5G Mobile Internet slicing (PKI environment)	The user in 5G Vehicle Internet slicing (CLC environment)
Input: A message m, the private key sk_p of a PKI user and the public key PK_c of a CLC user .	Input: The ciphertext σ, the private key sk_c of the CLC user and the public key PK_p of the PKI user.
Output: A ciphertext $\sigma = (c, u, V)$	Output: The verification is successful or failed.
1. Choose $k \in Z_q^*$ 2. Do the following calculation: 2.1. $R_1 = kP, h = H_2(m, R_1), R_2 = hP$ 2.2. $c = m \oplus H_3(R_2)$ 2.3. $u = (h - k)sk_p$ 2.4. $V = kPK_{c1} + T + \gamma P_{pub}$ 3. Deliver the ciphertext σ to the CLC user.	1. Do the following calculation: 1.1. $R_1 = \dfrac{1}{x_c}(V - dP), R_2 = R_1 + uPK_p$ 1.2. $m = c \oplus H_3(R_2)$ 1.3. $h = H_2(m, R_1)$ 2. Verify if $R_1 = hP - uPK_p$ 3. If the above equation holds, the verification is successful, otherwise return $\perp$ to indicate the failure.

FIGURE 2. The PCHS scheme.

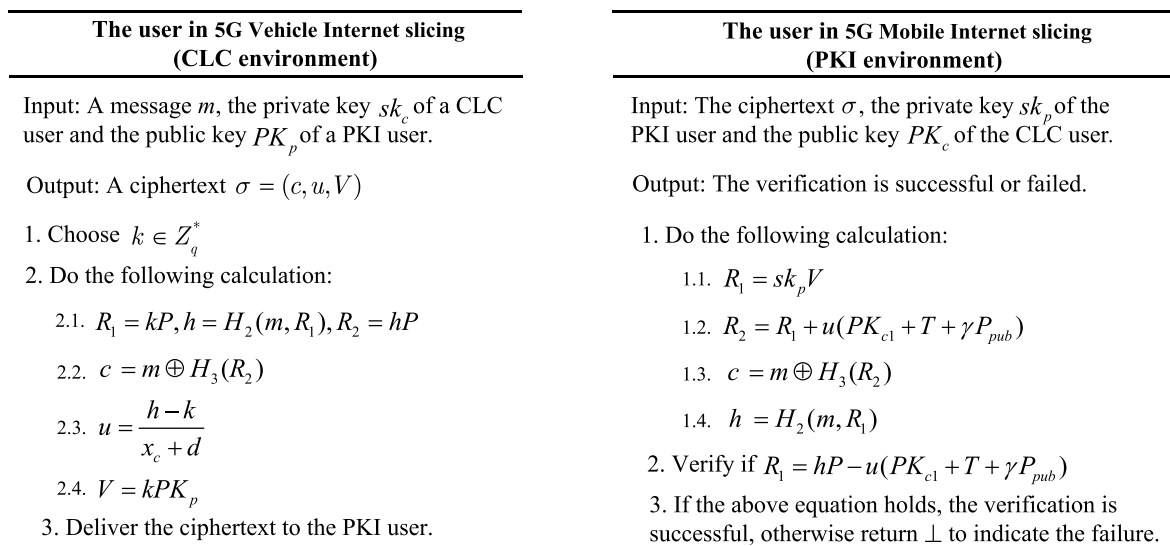| The user in 5G Vehicle Internet slicing (CLC environment)	The user in 5G Mobile Internet slicing (PKI environment)
Input: A message m, the private key sk_c of a CLC user and the public key PK_p of a PKI user.	Input: The ciphertext σ, the private key sk_p of the PKI user and the public key PK_c of the CLC user.
Output: A ciphertext $\sigma = (c, u, V)$	Output: The verification is successful or failed.
1. Choose $k \in Z_q^*$ 2. Do the following calculation: 2.1. $R_1 = kP, h = H_2(m, R_1), R_2 = hP$ 2.2. $c = m \oplus H_3(R_2)$ 2.3. $u = \dfrac{h - k}{x_c + d}$ 2.4. $V = kPK_p$ 3. Deliver the ciphertext to the PKI user.	1. Do the following calculation: 1.1. $R_1 = sk_p V$ 1.2. $R_2 = R_1 + u(PK_{c1} + T + \gamma P_{pub})$ 1.3. $c = m \oplus H_3(R_2)$ 1.4. $h = H_2(m, R_1)$ 2. Verify if $R_1 = hP - u(PK_{c1} + T + \gamma P_{pub})$ 3. If the above equation holds, the verification is successful, otherwise return $\perp$ to indicate the failure.

FIGURE 3. The CPHS scheme.

B. COMPLEXITY ASSUMPTIONS

The security of our schemes relies on the hardness of the following problems.

G_1 is a cyclic additive group of a large prime order q. P is a generator of G_1.

Definition 1: Computational Diffie-Hellman Problem (CDHP): Given an instance $(P, aP, bP) \in G_1$, for any $a, b \in Z_q^*$, it is difficult to compute $abP \in G_1$.

Definition 2: Discrete Logarithm Problem (DLP): Given an instance $(P, aP) \in G_1$, it is difficult to compute the integer $a \in Z_q^*$.

We assume there are two types of adversaries with different capabilities. The Type I adversary can replace users' public keys, but it does not know the master secret key s. The Type II adversary can access the master secret key s, but can not replace users' public keys.

III. PROPOSED SCHEMES

In this section, we assume that a 5G Mobile Internet slicing is in PKI public cryptosystem, and a 5G Vehicle Internet slicing is in CLC public cryptosystem. We propose two efficient signcryption schemes for the security authentication between the two heterogeneous 5G slicings. The first scheme PCHS allows users in PKI cryptosystem to signcrypt the messages and send them to users in CLC cryptosystem. Upon receiving these signcrypted messages, the users in CLC cryptosystem can decrypt and verify them. The second scheme CPHS is the inverse of PCHS. The detailed processes are shown in Fig. 2 and Fig. 3.

A. PCHS

The PCHS is described as follows:

Setup: Given a security parameter l, PKG chooses a cyclic additive group G_1 of a large prime order q. P is a

generator of G_1. Then, PKG defines three cryptographic hash functions: $H_1 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$, $H_2 : \{0,1\}^n \times G_1 \rightarrow Z_q^*$, and $H_3 : G_1 \rightarrow \{0,1\}^n$. PKG selects a master secret key $s \in Z_q^*$ randomly and computes the master public key $P_{pub} = sP$. Then, it publishes system parameters $\{G_1, P, P_{pub}, n, l, H_1, H_2, H_3\}$ and keeps the master key s secret.

PKI-KG: A user in PKI cryptosystem chooses a random number $x_p \in Z_q^*$ as his/her private key sk_p and computes $\frac{1}{x_p}P$ as his/her public key PK_p.

CLC-KG:

- Partial Private Key Extract: PKG randomly selects $t \in Z_q^*$ and computes $T = tP$, $\gamma = H_1(ID, T)$, and $d = t + s\gamma \in Z_q^*$. Then, PKG sends d and T to the user securely. d is the user's partial private key.
- Set Secret Value: The user randomly chooses $x_c \in Z_q^*$ as his/her secret value.
- Private Key Extract: The user sets his/her full private key as $sk_c = (x_c, d)$.
- Public Key Extract: The user sets his/her public key as $PK_c = (T, PK_{c1} = x_c P)$.

Signcrypt: The user in PKI cryptosystem uses his/her private key sk_p and the receiver's public key PK_c in CLC cryptosystem to signcrypt a message m as follows:

1) Choose a number $k \in Z_q^*$ randomly.
2) Compute $R_1 = kP$, $h = H_2(m, R_1)$, $R_2 = hP$.
3) Compute $c = m \oplus H_3(R_2)$.
4) Compute $u = (h - k)sk_p$.
5) Compute $V = kPK_{c1} + T + \gamma P_{pub}$.

The ciphertext is $\sigma = (c, u, V)$.

Unsigncrypt: The user in CLC cryptosystem uses his/her private key sk_c and the sender's public key PK_p in PKI cryptosystem to unsigncrypt the ciphertext $\sigma = (c, u, V)$ as follows:

1) Compute $R_1 = \frac{1}{x_c}(V - dP)$.
2) Compute $R_2 = R_1 + uPK_p$.
3) Compute $m = c \oplus H_3(R_2)$.
4) Compute $h = H_2(m, R_1)$.
5) Accept the message if and only if $R_1 = hP - uPK_p$, return $\perp$ otherwise.

Now we verify the correctness of the PCHS.
Firstly:

$$\frac{1}{x_c}(V - dP) = \frac{1}{x_c}(kPK_{c1} + T + \gamma P_{pub} - dP)$$
$$= \frac{1}{x_c}(kx_c P + tP + \gamma sP - dP)$$
$$= \frac{1}{x_c}(kx_c P + dP - dP)$$
$$= \frac{1}{x_c} \cdot kx_c P$$
$$= kP = R_1$$

Secondly:

$$R_1 + uPK_p = R_1 + (h - k)sk_p \cdot \frac{1}{x_p}P$$
$$= kP + (h - k)x_p \cdot \frac{1}{x_p}P$$
$$= kP + hP - kP$$
$$= hP = R_2$$

Finally:

$$hP - uPK_p = hP - (h - k)x_p \cdot \frac{1}{x_p}P$$
$$= hP - hP + kP$$
$$= kP = R_1$$

B. CPHS

In CPHS, the *Setup*, *PKI-KG* and *CLC-KG* algorithms are the same as PCHS. And CPHS can be described as follows:

Signcrypt: The user in CLC cryptosystem uses his/her private key sk_c and the receiver's public key PK_p in PKI cryptosystem to signcrypt a message m as follows:

1) Choose a number $k \in Z_q^*$ randomly.
2) Compute $R_1 = kP$, $h = H_2(m, R_1)$, $R_2 = hP$.
3) Compute $c = m \oplus H_3(R_2)$.
4) Compute $u = \frac{h-k}{x_c+d}$.
5) Compute $V = kPK_p$.

The ciphertext is $\sigma = (c, u, V)$.

Unsigncrypt: The user in PKI cryptosystem uses his/her private key sk_p and the sender's public key PK_c in CLC cryptosystem to unsigncrypt the ciphertext $\sigma = (c, u, V)$ as follows:

1) Compute $R_1 = sk_p V$.
2) Compute $R_2 = R_1 + u(PK_{c1} + T + \gamma P_{pub})$.
3) Compute $m = c \oplus H_3(R_2)$.
4) Compute $h = H_2(m, R_1)$.
5) Accept the message if and only if $R_1 = hP - u(PK_{c1} + T + \gamma P_{pub})$, return $\perp$ otherwise.

Now we verify the correctness of the CPHS.
Firstly:

$$sk_p V = ksk_p PK_p$$
$$= k\frac{1}{x_p} \cdot x_p P$$
$$= kP = R_1$$

Secondly:

$$R_1 + u(PK_{c1} + T + \gamma P_{pub})$$
$$= kP + \frac{h-k}{x_c+d} \cdot (PK_{c1} + T + \gamma P_{pub})$$
$$= kP + \frac{h-k}{x_c+d} \cdot (x_c P + tP + \gamma sP)$$
$$= kP + \frac{h-k}{x_c+d} \cdot (x_c + d)P$$
$$= kP + (h - k)P$$
$$= hP = R_2$$

Finally:

$$hP - u(PK_{c1} + T + \gamma P_{pub})$$
$$= hP - \frac{h-k}{x_c + d} \cdot (PK_{c1} + T + \gamma P_{pub})$$
$$= hP - \frac{h-k}{x_c + d} \cdot (x_c P + tP + \gamma sP)$$
$$= hP - \frac{h-k}{x_c + d} \cdot (x_c + d)P$$
$$= hP - (h - k)P$$
$$= kP = R_1$$

IV. THE SECURITY PROOF OF SCHEMES

In this section, we prove the security of the proposed schemes.

Theorem 1 (PKI-CLC IND-CCA2-1): In the random oracle model [28], if an adversary $\mathcal{A}_I$ has a nonnegligible advantage ξ against the IND-CCA2-1 security of the PCHS when performing q_i queries to oracles H_i ($i = 1, 2, 3$), there exists an algorithm through which the challenger $\mathcal{C}$ can solve the CDHP with an advantage ξ^*. Here, $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{e1c}}{q_1})$ $(1 - \frac{q_{e2c}}{q_1})(1 - \frac{q_u}{2^l})$, q_{e1c} denotes partial private key queries of CLC system, q_{e2c} denotes private key queries of CLC system, and q_u denotes unsigncryption queries.

Proof: To solve a random CDHP instance (P, aP, bP), $\mathcal{C}$ uses $\mathcal{A}_I$ as a subroutine. $\mathcal{A}_I$ should ask for H_1 before the identity ID_i is used in any other queries.

Initialization: $\mathcal{C}$ initializes and returns the system parameters $\{G_1, P, P_{pub}, n, l, H_1, H_2, H_3\}$ to $\mathcal{A}_I$. Next, it picks a challenged identity $\alpha \in \{1, 2, ..., q_1\}$ randomly without leaking anything about ID_α to $\mathcal{A}_I$. $\mathcal{C}$ needs to maintain the list of $L_1 \sim L_3$, LK_p and LK_c that are used to simulate the $H_1 \sim H_3$ hash oracles and the key extraction oracles respectively. Then, it answers these queries as follows.

Phase 1: $\mathcal{A}_I$ performs a polynomially bounded number of queries in an adaptive manner.

1. *H_1-Queries:* When $\mathcal{A}_I$ presents this query on an identity ID_i, $\mathcal{C}$ checks whether the tuple (ID_i, T_i, γ_i) exists in L_1. If so, $\mathcal{C}$ returns γ_i to $\mathcal{A}_I$. Otherwise, the public key query on ID_i is made to generate γ_i to $\mathcal{A}_I$ subsequently.

2. *H_2-Queries:* For a H_2 query, $\mathcal{C}$ first checks if (m_i, R_{1i}, h_i) has been in L_2 previously. If so, $\mathcal{C}$ returns the value h_i. Otherwise, $\mathcal{C}$ returns a random $h_i \in Z_q^*$, and adds new tuple (m_i, R_{1i}, h_i) to L_2.

3. *H_3-Queries:* For a H_3 query, $\mathcal{C}$ first checks if the value of H_3 query has been in L_3 previously. If so, $\mathcal{C}$ returns it. Otherwise, $\mathcal{C}$ randomly chooses h_{3i} from $\{0, 1\}^n$, returns h_{3i} and adds (R_{2i}, h_{3i}) to L_3.

4. *PKI Private-Key-Queries:* For a private key query on ID_i, $\mathcal{C}$ will invoke LK_p and search (ID_i, x_{pi}, PK_{pi}), then returns the private key x_{pi}.

5. *CLC Partial-Private-Key-Queries:* For a partial private key query on ID_i, $\mathcal{C}$ makes the following response:
   (1) If $ID_i = ID_\alpha$, $\mathcal{C}$ aborts.
   (2) If $ID_i \neq ID_\alpha$, $\mathcal{C}$ will invoke LK_c and search $(ID_i, x_{ci}, d_i, T_i, PK_{c1i})$, then return the partial private key d_i.

6. *CLC Private-Key-Queries:* For a private key query above ID_i, $\mathcal{C}$ responds as follows:
   (1) If $ID_i = ID_\alpha$, $\mathcal{C}$ aborts.
   (2) If $ID_i \neq ID_\alpha$, $\mathcal{C}$ will invoke LK_c and search $(ID_i, x_{ci}, d_i, T_i, PK_{c1i})$, then return (x_{ci}, d_i).

7. *CLC Public-Key-Queries:* When $\mathcal{A}_I$ submits a query on identity ID_i for his/her public key, $\mathcal{C}$ searches LK_c for the tuple $(ID_i, x_{ci}, d_i, T_i, PK_{c1i})$ and responds as follows:
   (1) If the public key exists, $\mathcal{C}$ returns (T_i, PK_{c1i}).
   (2) If $ID_i = ID_\alpha$, $\mathcal{C}$ randomly chooses $x_{c\alpha}, \gamma_\alpha \in Z_q^*$, $bP \in G_1$ with unknown b, then it sets $PK_{c1\alpha} = bP$, $T_\alpha = d_{i\alpha}P - \gamma_\alpha P_{pub}$, where $\gamma_\alpha = H_1(ID_\alpha, T_\alpha)$. $\mathcal{C}$ updates $(ID_\alpha, x_{c\alpha}, d_{i\alpha}, T_\alpha, PK_{c1\alpha})$ in LK_c and $(ID_\alpha, T_\alpha, \gamma_\alpha)$ in L_1.
   (3) If $ID_i \neq ID_\alpha$, $\mathcal{C}$ selects $\gamma_i, x_{ci}, d_i \in Z_q^*$ randomly. Next, $\mathcal{C}$ calculates $PK_{c1i} = bP$, $T_i = d_iP - \gamma_i P_{pub}$, where $\gamma_i = H_1(ID_i, T_i)$. Then, $\mathcal{C}$ updates LK_c with $(ID_i, x_{c,i}, d_i, T_i, PK_{c1i})$ and L_1 with (ID_i, T_i, γ_i).

8. *CLC Public-Key-Replacement-Queries:* When $\mathcal{A}_I$ replaces the public key PK_{ci} of the identity ID_i with PK_{ci}^*, $\mathcal{C}$ updates LK_c with the tuple $(ID_i, \emptyset, \emptyset, T_i, PK_{ci}^*)$. Here, $\emptyset$ denotes an unknown value. The new public key is used by the challenger to solve the CDHP or is requested by the adversary in the public key queries.

9. *Signcrypt-Queries:* Suppose ID_{send} and $ID_{receive}$ are the identities of a sender and a receiver respectively. When $\mathcal{A}_I$ makes this query on the tuple $(\sigma, ID_{send}, ID_{receive})$, $\mathcal{C}$ makes responses as below:
   (1) If $ID_{send} \neq ID_\alpha$, $\mathcal{C}$ runs the signcrypt algorithm normally and sends the ciphertext to $\mathcal{A}_I$.
   (2) If $ID_{send} = ID_\alpha$ and $ID_{receive} \neq ID_\alpha$, $\mathcal{C}$ finds $(d_\alpha, x_{c\alpha}, PK_p)$ from LK_c and LK_p and generates the ciphertext in following steps:
   - Choose $k \in Z_q^*$ randomly, compute $R_1 = kP$;
   - Compute $h = H_2(m, R_1)$, $R_2 = hP$, store (m, R_1, h) into L_2;
   - Compute $c = m + H_3(R_2)$;
   - Compute $u = \frac{h-k}{x_c + d}$, $V = kPK_p$;
   - The ciphertext is $\sigma = (c, u, V)$.

10. *Unsigncrypt-Queries:* Upon receiving a unsigncrypt query of $(\sigma, ID_{send}, ID_{receive})$, $\mathcal{C}$ responds as follows:
   (1) If $ID_{receive} \neq ID_\alpha$, $\mathcal{C}$ runs the unsigncrypt algorithm normally and returns the result.
   (2) If $ID_{receive} = ID_\alpha$, $\mathcal{C}$ searches L_3 for h_3. $\mathcal{C}$ computes $m = c \oplus h_3$. If and only if $R_1 = hP - uPK_p$, the message is accepted. Otherwise, the ciphertext is rejected.

Challenge phase: $\mathcal{A}_I$ generates two equal-length plaintexts m_0, m_1 and two challenged identities ID_{send}^*, $ID_{receive}^*$. In phase 1, the public key of $ID_{receive}^*$ can not be replaced and the partial private key queries can not be asked as well as the secret value. If $ID_{receive}^* \neq ID_\alpha$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ asks public key request oracle on ID_{send}^*, sets receiver's partial public key to bP (an instance of CDHP) for an unknown b. Next, $\mathcal{C}$ randomly selects $T \in G_1$, $k^* \in Z_q^*$, $\gamma^*, h_3^* \in \{0, 1\}^n$ and a random bit $\mu \in \{0, 1\}$, then sets $R_1^* = aP$, $PK_{c1}^* = bP$, $c^* = m_\mu \oplus h_3^*$, computes $V^* = k^*PK_{c1}^* + T + \gamma^*P_{pub}$ and returns (c^*, u^*, V^*) to $\mathcal{A}_I$.

Phase 2: As in Phase 1, $\mathcal{A}_I$ will also present adaptively queries with the limitations of Type I adversary in this phase. However, (1) $\mathcal{A}_I$ cannot submit the private key query on $ID_{receive}^*$. (2) $\mathcal{A}_I$ cannot ask the partial private key query on ID_{send}^* if the public key is replaced before the challenge. (3) $\mathcal{A}_I$ cannot present the unsigncrypt query on $(\sigma^*, ID_{send}^*, ID_{receive}^*)$.

Guess: In order to get a correct guess, $\mathcal{A}_I$ should obtain the outputs $R_1^* = aP$, $PK_{c1}^* = bP$ and $V^* = k^*PK_{c1}^* + T + \gamma^* P_{pub}$ from the challenge phase. Given an instance (P, aP, bP), $\mathcal{C}$ can solve CDHP: $abP = V^* - T - \gamma^* P_{pub}$.

Probability Analysis: In above discussions, there are four situations leading to the aborting of $\mathcal{C}$:

- E_1: $\mathcal{A}_I$ asks the partial private key queries of the challenged identity ID_α.
- E_2: $\mathcal{A}_I$ presents the private key queries of the challenged identity ID_α.
- E_3: $\mathcal{A}_I$ does not choose ID_α as the receiver's identity in the challenge phase.
- E_4: $\mathcal{C}$ aborts in an unsigncryption query because of rejecting a valid ciphertext.

Only if $\mathcal{C}$ does not reject the game, the CDHP can be solved. As a result, the probability of $\mathcal{C}$ not aborting is: $\Pr[\neg abort] = \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3 \wedge \neg E_4]$.

We know that $\Pr[E_1] = \frac{q_{e1c}}{q_1}$, $\Pr[E_2] = \frac{q_{e2c}}{q_1}$, $\Pr[E_3] = (1 - \frac{1}{q_1})$, and $\Pr[E_4] \leq \frac{q_u}{2^l}$. Therefore, $\Pr[\neg abort] \geq \frac{1}{q_1}(1 - \frac{q_{e1c}}{q_1})(1 - \frac{q_{e2c}}{q_1})(1 - \frac{q_u}{2^l})$.

In addition, the probability that $\mathcal{C}$ randomly chooses a T from L_1 and outputs it as a solution of CDHP is $\frac{1}{q_1}$. In conclusion, we have $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{e1c}}{q_1})(1 - \frac{q_{e2c}}{q_1})(1 - \frac{q_u}{2^l})$.

Theorem 2 (PKI-CLC IND-CCA2-2): In the random oracle model, if an adversary $\mathcal{A}_{II}$ has a nonnegligible advantage ξ against the IND-CCA2-2 security of the PCHS when performing q_i queries to oracles H_i ($i = 1, 2, 3$), there exists an algorithm through which the challenger $\mathcal{C}$ can solve the CDHP with an advantage ξ^*. Here, $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{e2c}}{2^l})(1 - \frac{q_u}{2^l})$.

The $H_1 \sim H_3$ hash oracles and the key extraction oracles are similar with PKI-CLC IND-CCA2-1. The second type adversary $\mathcal{A}_{II}$ knows the master key of PKG, but it is not allowed to replace any user's public key.

Probability Analysis: In Phase 2, $\mathcal{A}_{II}$ can not ask a private key query on $ID_{receive}^*$ or an unsigncryption query on σ^* for the corresponding plaintext. $\mathcal{C}$ will abort the game in following situations:

- E_1: $\mathcal{A}_{II}$ asks the private key queries of the challenged identity ID_α.
- E_2: $\mathcal{A}_{II}$ does not choose ID_α as the receiver's identity in the challenge phase.
- E_3: $\mathcal{C}$ aborts in an unsigncryption query because of rejecting a valid ciphertext.

Only if $\mathcal{C}$ does not reject the game, the CDHP can be solved. As a result, the probability of $\mathcal{C}$ not aborting is: $\Pr[\neg abort] = \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3]$.

We know that $\Pr[E_1] = \frac{q_{e2c}}{q_1}$, $\Pr[E_2] = (1 - \frac{1}{q_1})$, and $\Pr[E_3] \leq \frac{q_u}{2^l}$. Therefore, $\Pr[\neg abort] \geq \frac{1}{q_1}(1 - \frac{q_{e2c}}{q_1})(1 - \frac{q_u}{2^l})$.

In addition, the probability that $\mathcal{C}$ randomly chooses a T from L_1 and outputs it as a solution of CDHP is $\frac{1}{q_1}$. In conclusion, we have $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{e2c}}{q_1})(1 - \frac{q_u}{2^l})$.

Theorem 3 (PKI-CLC EUF-CMA): In the random oracle model, if an adversary $\mathcal{F}$ has a nonnegligible advantage ξ against the EUF-CMA security of the PCHS when performing q_i queries to oracles H_i ($i = 1, 2, 3$), then there exists an algorithm through which the challenger $\mathcal{C}$ can solve the DLP with an advantage ξ^*. Here, $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{ep}}{2^l})(1 - \frac{q_s}{2^l})$, q_{ep} denotes private key queries of PKI system and q_s denotes signcryption queries.

Proof: To solve a random DLP instance (P, aP), $\mathcal{C}$ uses $\mathcal{F}$ as a subroutine. $\mathcal{F}$ should ask for H_1 before the identity ID_i is used in any other queries.

Initialization: $\mathcal{C}$ initializes and returns the system parameters $\{G_1, P, P_{pub}, n, l, H_1, H_2, H_3\}$ to $\mathcal{F}$. Next, it picks a challenged identity $\alpha \in \{1, 2, ..., q_1\}$ randomly without leaking anything about ID_α to $\mathcal{F}$. $\mathcal{C}$ needs to maintain the list of $L_1 \sim L_3$, LK_p and LK_c that are used to simulate the $H_1 \sim H_3$ hash oracles and the key extraction oracles respectively.

Training: $\mathcal{F}$ performs a polynomially bounded number of queries in an adaptive manner. The queries in this phase are the same as the queries described in Theorem 1.

Forgery: After the training, $\mathcal{F}$ outputs a forgery $(\sigma^*, ID_{send}^*, ID_{receive}^*)$. During the training, $\mathcal{F}$ cannot make a private key query on ID_{send}^*. If $ID_{send}^* \neq ID_\alpha$, $\mathcal{C}$ aborts. If $ID_{send}^* = ID_\alpha$, $\mathcal{C}$ invokes the L_1 and LK_c to search γ^*, T^*, PK_{c1}^* and R_1. Then, $\mathcal{C}$ obtains $aPK_{c1} = V - T - \gamma P_{pub}$ and $aPK_{c1}^* = V - T^* - \gamma^* P_{pub}$. Finally, $\mathcal{C}$ outputs $a = \frac{[(t-t^*)+s(\gamma-\gamma^*)]}{(x_c-x_c^*)}$ as the solution of DLP. The proof is as follows:

$$aPK_{c1} - aPK_{c1}^*$$
$$= (V - T - \gamma P_{pub}) - (V - T^* - \gamma^* P_{pub})$$
$$= (T^* - T) + (\gamma^* - \gamma)P_{pub}$$

Probability Analysis: For above discussions, there are three situations leading to the aborting of $\mathcal{C}$:

- E_1: $\mathcal{F}$ asks the private key queries of the challenged identity ID_α
- E_2: $\mathcal{F}$ does not choose ID_α as the sender's identity in the challenge phase.
- E_3: $\mathcal{C}$ aborts in a signcryption query due to the collision on hash operation.

Only if $\mathcal{C}$ does not reject the game, the DLP can be solved. As a result, the probability of $\mathcal{C}$ not aborting is: $\Pr[\neg abort] = \Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3]$.

We know that $\Pr[E_1] = \frac{q_{ep}}{q_1}$, $\Pr[E_2] = (1 - \frac{1}{q_1})$, and $\Pr[E_3] \leq \frac{q_s}{2^l}$. Therefore, $\Pr[abort] \geq \frac{1}{q_1}(1 - \frac{q_{ep}}{q_1})(1 - \frac{q_s}{2^l})$.

In addition, the probability that $\mathcal{F}$ correctly guesses the hash value of H_1 is $\frac{1}{q_1}$. In conclusion, we have $\xi^* \geq \frac{\xi}{q_1^2}(1 - \frac{q_{ep}}{q_1})(1 - \frac{q_s}{2^l})$.

The security proof of CPHS is similar to that of PCHS.

TABLE 1. Performance comparison of each signature scheme.

The scheme	Key Generation	Signcryption	Unsigncryption	Communication cost	Communication Direction
LZT-I[24]	4S+H	P+3S+3H+E	3P+2H+E	$2|G_1|+|m|$	PKI $\rightarrow$ IBC
LZT-II[24]	4S+H	P+2S+2H+E	3P+S+3H+E	$2|G_1|+|m|$	IBC $\rightarrow$ PKI
LHJ[25]	4S+2H	P+3S+3H+E	5P+3H	$2|G_1|+|m|$	CLC $\rightarrow$ IBC
ZZW[26]	P+5S+2H	P+3S+2H+2E	2P+2S+3H+E	$3|G_1|+|ID|+|m|$	IBC $\rightarrow$ CLC
WLZ[27]	2P+7E	3S+H+4E	5P+2S+H+E	$3|G_1|+|G_2|$	IBC $\rightarrow$ PKI
PCHS	4S+H	4S+2H	3S+2H	$|G_1|+|Z_q^*|+|m|$	PKI $\rightarrow$ CLC
CPHS	4S+H	3S+2H	4S+2H	$|G_1|+|Z_q^*|+|m|$	CLC $\rightarrow$ PKI

V. PERFORMANCE ANALYSIS

In this section, we assume that the client is in 5G Mobile Internet slicing (PKI environment) and the server is in 5G Vehicle Internet slicing (CLC environment). To simulate the inter-process communication between the client and the server in different public cryptosystems. We set up two Raspberry Pis as the experimental platform. The structure diagram of the interprocess communication is shown in Fig. 4. and the simulation results are shown in Fig. 5.
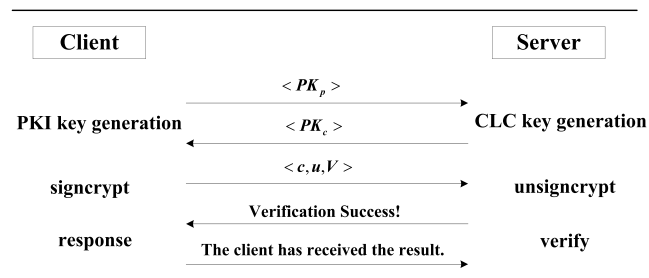
FIGURE 4. The structure diagram of the interprocess communication.

FIGURE 5. The simulation result between the client and the server. (a) The client. (b) The server.

In the first step, the client generates the public key PK_p and sends it to the server. Also the server receives his/her public key PK_c to the client. In the second step, via the signcryption, the client generates the ciphertext σ and sends it to the server. Then, the server unsigncrypts and verifies the ciphertext. If it is verified, the server sends "Verification Success!" to the client. In the last step, if the client obtains the response from the server, it sends "The client has received the result." to the server as a reply.

For the theoretical complexity analysis, we compare our schemes with several existing schemes [24]–[27] in Table 1.

Let $|G_1|$ and $|G_2|$ denote the length of the elements in group G_1 and G_2 respectively, $\left|Z_q^*\right|$ denote the length of an element in field Z_q^*, $|m|$ denote the length of a message, $|ID|$ denote the the length of a user's identity. S is the scalar multiplication in G_1, E means the exponentiation in G_2, P denotes the bilinear pairing, H is the hash operation in Z_q^*.

From Table 1, we can clearly find that all schemes except ours use bilinear pairings and exponentiation computations in the signcryption and unsigncryption stages. Furthermore, compared with the schemes in [24]–[26], the scheme [27] involves more bilinear pairings, which lead to more time consumption. As for the communication overhead, WLZ [27] is the highest among these schemes because one of its ciphertext elements is in group G_2. Observing the last column in Table 1, we find that the schemes in [25]–[27] are one-way communications, which have the limitations in practice. The schemes in [24]–[27] are only suit for the heterogeneous environments of PKI-to-IBC or IBC-to-CLC. Our schemes are designed specially for the mutual heterogeneous communication of PKI-to-CLC public cryptosystems innovatively.

TABLE 2. Operating time in different simulation environments(ms).

Environment \ Operation	P	S	E	H
Ubuntu OS	1.396	1.812	0.197	0.001
Raspberry Pi	15.104	10.006	1.955	0.002

For the quantitative analysis, we set up two different environments for simulation. The first experimental platform is in Ubuntu OS over an Inter Pentium 2.70 GHz dual core processor and 1024 MB memory. The second one is in Raspberry Pi 3B+. The cryptography library we used is PBC-0.5.14. We choose the type A curve $y^2 = x^3 + x$ in simulation because the math algorithm is more efficient on exponentiation in G_2. We first assess the cryptographic operations, and each operation is run 10000 times to eliminate the influence of random disturbance. The detailed results are shown in Table 2. Generally, the time overhead in Raspberry Pi is about ten times higher than that in Ubuntu. Furthermore, we also find that the time cost on the scalar multiplication is higher than that on the bilinear pairing in Ubuntu environment, while the result is opposite in Raspberry Pi platform, which would lead to the significant changes among schemes in the two simulation environments.
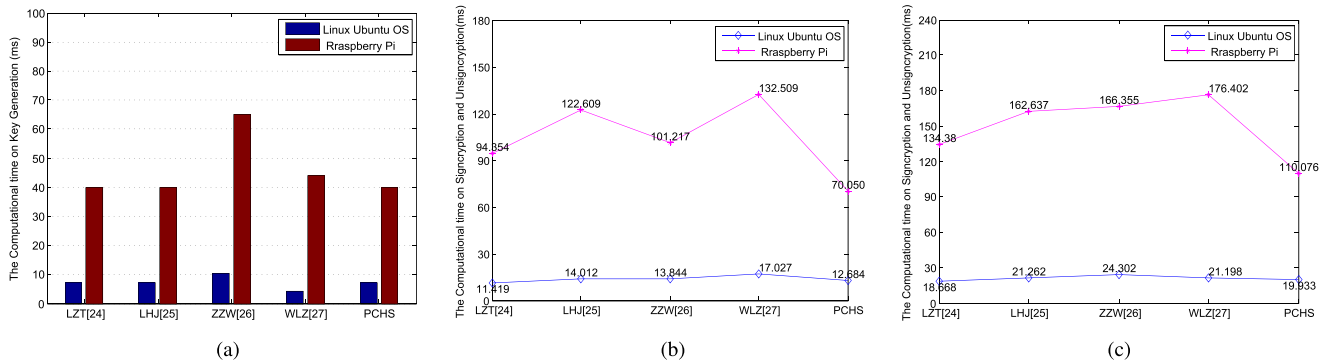
FIGURE 6. Comparison of time consumption among different schemes. (a) Time comparison on key generation. (b) Time comparison on signcryption and unsigncryption. (c) Time comparison in total.

Before running the schemes, we first test the time consumption on their private/public key pair generation. As shown in Fig. 6(a), although the time cost on key generation of each scheme is very close, WLZ [27] achieves the lowest computation cost in this part because there is no scalar multiplication but only a few bilinear pairings. Fig. 6(b) shows the time consumption on the signcryption and unsigncryption among different schemes in two platforms. Fig. 6(c) indicates the total overhead of the key generation, the signcryption and the unsigncryption. As a whole, ZZW [26] has the highest computational overhead in Ubuntu environment, while WLZ [27] takes the highest cost in Raspberry Pi platform. Without any bilinear pairing, our scheme is more efficient on the computational overhead.

VI. CONCLUSION

In this paper, we propose two signcryption schemes without the bilinear pairing between PKI and CLC cryptosystems, which can achieve secure mutual heterogeneous communications of 5G network slicings. When the users in the 5G slicing based on PKI environment (such as a Mobile Internet slicing) try to communicate with the users in the 5G slicing based on CLC (such as a Vehicle Internet slicing), our schemes can provide the available and efficient solutions. Meanwhile, they have the IND-CCA2 under the CDHP and the EUF-CMA under the DLP in the random oracle model. We also simulate the interprocess communication between two different public cryptosystems on Raspberry Pi. Compared with the existing schemes, our schemes are innovative and more efficient for heterogeneous communications of 5G network slicings.

REFERENCES

[1] A. Osseiran *et al.*, "Scenarios for 5G mobile and wireless communications: The vision of the METIS project," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.

[2] Y. G. Ren and L. Zhang, "Prospect of the fifth generation mobile communication system," *Inf. Commun.*, no. 8, pp. 255–256, 2014.

[3] T. Wen and P. Y. Zhu. (2013). *5G: A Technology Vision*. [Online]. Available: http://www.huawei.com/en/about-huawei/publications/winwin-magazine/hw-329304.htm

[4] H. Li and Y. L. Fu, "Analysis and prospect on 5G network security," *Radio Commun. Technol.*, vol. 41, no. 4, pp. 1–7, 2015.

[5] X. Yao, X. Han, X. Du, and X. Zhou, "A lightweight multicast authentication mechanism for small scale IoT applications," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3693–3701, Oct. 2013.

[6] M. Iwamura, "NGMN view On 5G architecture," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., May 2015, pp. 1–5.

[7] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.

[8] X. H. You, Z. W. Pan, and X. Q. Gao, "The 5G mobile communication: The development trends and its emerging key techniques," *Sci. China*, vol. 44, no. 5, pp. 551–563, 2014.

[9] Y. Xu, G. Y. Gao and L. Wang, "Initial analysis of 5G mobile network slicing technology," *Des. Techn. Posts Telecommun.*, no. 7, pp. 19–22, 2016.

[10] H. Hu and X. Fei, "Architecture design and standardization progress of 5G network," *Telecommun. Sci.*, no. 4, pp. 126–132, 2016.

[11] A. Shamir, "Identity-based cryptosystem and signature scheme," in *Proc. Adv. Cryptol. (CRYPTO)*, vol. 196. 1985, pp. 47–53.

[12] Y. Xiao *et al.*, "A survey of key management schemes in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.

[13] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.

[14] S. S. Al-riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Adv. Cryptol.*, 2003, pp. 452–473.

[15] Y. Zhang, "Digital signcryption or how to achieve cost(signature & encryption) cost(signature) + cost(encryption)," in *Proc. Cryptol.*, 1997, pp. 165–179.

[16] J. Baek, R. Steinfeld, and Y. L. Zhang, "Formal proofs for the security of signcryption," in *Proc. Cryptol. PKC*, Paris, France, 2002, pp. 81–98.

[17] C. K. Li, G. Yang, and D. S. Wong, "An efficient signcryption scheme with key privacy," in *Proc. 4th Eur. Public Key Infrastructure: Theory Pract. EuroPKI*, Palma de Mallorca, Spain, 2007, pp. 78–93.

[18] A. Yin and H. Liang, "Certificateless hybrid signcryption scheme for secure sommunication of wireless sensor networks," *Wireless Pers. Commun.*, vol. 80, no. 3, pp. 1049–1062, 2015.

[19] X. Du, M. Guizani, Y. Xiao, and H. H. Chen, "Secure and efficient time synchronization in heterogeneous sensor networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2387–2394, Jul. 2008.

[20] X. Du, Y. Xiao, H. H. Chen, and Q. Wu, "Secure cell relay routing protocol for sensor networks," *Wireless Commun. Mobile Comput.*, vol. 6, no. 3, pp. 375–391, May 2006.

[21] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[22] Y. X. Sun and H. Li, "Efficient signcryption between TPKC and IDPKC and its multi-receiver construction," *Sci. China*, vol. 53, no. 3, pp. 557–566, 2010.

[23] Q. Huang, D. S. Wong, and G. Yang, "Heterogeneous signcryption with key privacy," *Comput. J.*, vol. 54, no. 4, pp. 525–536, 2011.

[24] F. Li, H. Zhang, and T. Takagi, "Efficient signcryption for heterogeneous systems," *IEEE Syst. J.*, vol. 7, no. 3, pp. 420–429, Sep. 2013.

[25] F. Li, Y. Han, and C. Jin, "Practical access control for sensor networks in the context of the Internet of Things," *Comput. Commun.*, vols. 89–90, pp. 154–164, Sep. 2016.

[26] Y. L. Zhang, L. G. Zhang, and C. F. Wang, "Provable secure IDPKC-to-CLPKC heterogeneous signcryption scheme," *J. Electron. Inf. Technol.*, vol. 39, no. 9, pp. 2127–2133, 2017.

[27] C. Wang, Y. Li, S. Niu, and Y. Zhang, "Efficient heterogeneous signcryption scheme in the standard model," *J. Electron. Inf. Technol.*, vol. 39, no. 4, pp. 881–886, 2017.

[28] A. A. Omala, N. Robert, and F. G. Li, "A provably-secure transmission scheme for wireless body area networks," *J. Med. Syst.*, vol. 40, no. 11, pp. 247–261, 2016.

JINGWEI LIU (M'11) received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in communication and information systems from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. He has published over 40 papers in journals and conference proceedings. His research interests include information security, network security, and cryptography. He is a member of the Chinese Association for Cryptologic Research.

LIHUAN ZHANG is currently pursuing the degree with the School of Telecommunications Engineering, Xidian University. She has published two journal papers. Her research interests include secure authentication protocols and signcryption algorithm.
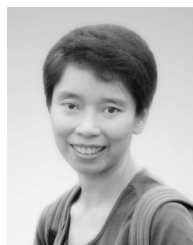
RONG SUN (M'10) received the B.E. degree in telecommunications engineering and the M.E. and Ph.D. degrees in communications and information systems from Xidian University, Xi'an, China, in 1998, 2001, and 2008, respectively. She is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. Her research interests include wireless communications, channel coding design, and information theory. She is a member of IEICE.

XIAOJIANG (JAMES) DU (M'03–SM'09) received the B.E. degree from Tsinghua University, China, in 1996, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 2002 and 2003, respectively, all in electrical engineering. He is currently a Professor with the Department of Computer and Information Sciences, Temple University. His research interests are security, systems, wireless networks, and computer networks. He has published over 240 journal and conference papers in these areas. He serves on the editorial boards of two international journals.

MOHSEN GUIZANI (S'85–M'89–SM'99–F'09) received the bachelor's (Hons.) and master's degrees in electrical engineering and the master's and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor and the ECE Department Chair with the University of Idaho. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the TAOS Technical Committee. He currently serves on the editorial boards of several international technical journals. He is the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley). He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005.

● ● ●