Multi-Heterogeneous Signcryption Scheme for Next Generation Slicing Networking

Bei Gong, Yong Wu, Jiangjiang Zhang, Shanshan Tu, *Member, IEEE,*
Hisham Alasmary, *Member, IEEE,* Muhammad Waqas, *Senior Member, IEEE*

***Abstract*—With the advent of 5G and the forthcoming 6G networks, network slicing has emerged as a pivotal technology to address the diverse and dynamic communication needs of modern systems. It enables the creation of multiple virtual networks over a shared physical infrastructure, optimizing resource utilization and providing tailored services. However, due to the security and service requirements of different slices, different network slicings usually adopt different public key cryptosystems, that is, heterogeneous network slicing, which inevitably leads to security challenges in data transmission of heterogeneous network devices. To address this, this paper proposes a generalized multi-heterogeneous signcryption scheme (MHSC) for network slicing, which ensures secure and efficient data transmission among devices utilizing three widely adopted public key cryptography systems: Public Key Infrastructure (PKI), Identity-Based Cryptography (IBC), and Certificateless Cryptography (CLC). Based mathematically hard problems, MHSC achieves multiple security properties, including forward secrecy and public verifiability, ensuring a higher level of security. In computational and communication efficiency, MHSC is evaluated through both theoretical analysis and simulation experiments. The results demonstrate that MHSC can improve computational efficiency by 17.66% and communication efficiency by 5.94% on average, offering a scalable and efficient solution for secure data transmission in heterogeneous network slicings.**

***Index Terms*—heterogeneous signcryption, generalized signcryption, random oracle model, network slicing.**

I. Introduction

With the rapid advancement of 5G and the development of 6G networks, network slicing has emerged as a fundamental technology for addressing the diverse and dynamic communication demands [1], [2]. It leverages software-defined networking and network function virtualization to logically partition network resources into isolated end-to-end slices, each customized to meet specific performance, security, and reliability requirements. Through dynamic resource allocation, these slices optimize utilization and adapt to fluctuating demands, ensuring efficient and flexible network services [3]. For instance, low-latency slices can support real-time data transmission for autonomous vehicles or telemedicine, while high-bandwidth slices are well-suited for high-throughput applications such as HD video streaming or virtual reality.

However, the widespread adoption of network slicing has also introduced significant security challenges. Since network slicing is designed to provide customized services, different slices may employ different cryptographys to meet their specific security requirements [4]. In high-security and identity-trust-demanding applications, such as smart cities and financial transactions, a certificate-based Public Key Infrastructure (PKI) is typically used [5]. PKI is responsible for distributing and managing public key certificates for devices to ensure identity trust and secure data transmission. However, as the number of devices grows, managing certificates becomes increasingly complex, leading to significant computational, storage, and communication overhead. In scenarios requiring comprehensive data monitoring, such as remote healthcare and military communications, an Identity-Based Cryptography (IBC) is commonly employed. In IBC, a device's identity serves as its public key, and the Private Key Generator (PKG) generates the corresponding private key based on the device's identity, enabling real-time decryption and monitoring of data within the slice. However, IBC presents the key escrow problem—if the PKG is compromised, the private keys of all devices within the slice could be exposed [6]. For resource-constrained and large-scale device environments, such as the Industrial Internet of Things (IIoT) and drone swarms, the Certificateless Cryptography (CLC) is often used [7]. CLC employs a Key Generation Center (KGC) to collaboratively generate keys with users, eliminating the need for certificate management and mitigating the key escrow issue. However, CLC offers lower identity authentication security compared to PKI and imposes higher security requirements on communication protocols. Therefore, due to the varying security characteristics and application scenarios of different public key cryptosystems, network slicing must adopt appropriate cryptosystems based on specific security needs, resulting in a heterogeneous environment [8].

Furthermore, existing applications are often not confined to a single network slicing but span across multiple slicings for complex data interactions. Specifically, in Intelligent Transportation Systems (ITS), autonomous vehicles, traffic management systems, and Roadside Units (RSUs) may reside in different network slicings. Each slicing utilizes a distinct cryptographic system to meet varying service quality requirements, and the slicing collaborate with each other to ensure the efficient operation of the entire ITS ecosystem. As a result,

B. Gong, Y. Wu and S. Tu are with the Faculty of Information Technology, Beijing University of Technology, 100124 Beijing, China, e-mail: (gongbei@bjut.edu.cn, weliars@emails.bjut.edu.cn, sstu@bjut.edu.cn).

J. Zhang is with the School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, 030006, China, e-mail: (jiangofyouth@sxu.edu.cn).

H. Alasmary is with the Department of Computer Science, King Khalid University, Abha, Saudi Arabia, e-mail: (alasmary@kku.edu.sa).

M. Waqas is with the Centre for Sustainable Cyber Security, Faculty of Engineering and Science, University of Greenwich, SE10 9LS, London, UK and with the School of Engineering, Edith Cowan University, Perth 6027, WA, Australia (e-mail: engr.waqas2079@gmail.com).

to enable interaction and information propagation between network slicings, cross-slice data transmission must ensure security in terms of data integrity and confidentiality. Since the key generation mechanisms and security requirements vary across heterogeneous cryptosystems, a new challenge arises: How can secure data transmission be ensured between network slicings that employ different public-key cryptosystems?

In addition, the diversity of devices in heterogeneous slicings imposes higher performance demands on data transmission. In particular, in resource-constrained application scenarios like IoT and Wireless Body Area Networks (WBANs), where devices typically rely on portable batteries with limited computation and communication capabilities, it becomes difficult to support complex encryption algorithms and authentication protocols [9], [10]. Thus, designing a secure, efficient, and flexible data transmission mechanism for heterogeneous slices is essential.

To secure data interactions between heterogeneous cryptosystems, researchers have proposed various heterogeneous signcryption schemes [11]. However, existing solutions only support one-way or mutual data interaction between any two of the PKI, IBC, and CLC cryptographic systems, failing to meet the need for secure and efficient interoperability among all three. Even when multiple heterogeneous signcryption schemes are used to enable secure data transmission across the three cryptographic systems, differences in key generation methods and the uniform ciphertext format across these schemes may lead to multiple public-private key pairs on the same device and ciphertext parsing confusion. This hinders efficient transmission between heterogeneous devices. Additionally, terminal devices may need to maintain multiple pairs of public and private keys, increasing the overall complexity of key management. More importantly, the security standards of various heterogeneous signcryption mechanisms lack uniformity. If a vulnerability exists in one mechanism, attackers may exploit it to spread malicious information, further propagating threats across the entire system through associated heterogeneous signcryption mechanisms, ultimately posing a severe risk to the overall security of the heterogeneous network.

Motivated by these considerations, we propose a generalized multi-heterogeneous signcryption scheme for network slicings, and the main contribution as follows.

1) *Generalized Heterogeneous Signcryption Scheme Proposed.* We propose a generalized multi-heterogeneous signcryption scheme for next generation slicing network, as MHSC. MHSC integrates a signcryption mechanism, enabling secure and efficient data transmission between heterogeneous devices across any combination of PKI, IBC, and CLC cryptosystems within network slices.
2) *Security Proofs and Analysis.* Based on Random Oracle Model (ROM), we proof the confidentiality and unforgeability of MHSC. Additionally, based on security assumptions, MHSC satisfies several security properties, including non-repudiation, integrity, public verifiability, forward secrecy, known session-specific temporary information security and scalability, ensuring security data transmission across heterogeneous network slices and showing the scalability of practical applications.
3) *Performance Evaluation and Efficiency Improvement.* Through both theoretical analysis and simulation experiments, MHSC outperforms existing solutions in computational and communication efficiency. Specifically, MHSC improves computational efficiency by approximately 17.66% and communication efficiency by 5.94% on average, making it better suited for heterogeneous network slicings.

The overall structure of this paper is as follows. In Section II, we analyze and discuss the research status related to our paper. In Section III, we organize the fundamental knowledge relevant to MHSC. In Section IV, we present the proposed generalized multi-heterogeneous signcryption scheme. In Section V, we conduct the security analysis of MHSC. In Section VI, we perform simulation experiments to compare the performance of MHSC. In Section VII, we provide a conclusion of the entire paper.

II. Related Work

With the accelerated deployment of 5G and the evolution toward 6G, network slicing has emerged as a pivotal technology in next-generation network architectures, enabling customized services over shared infrastructure [12]. To accommodate diverse security and service requirements, different slices frequently adopt heterogeneous public key cryptographic systems, such as PKI, IBC, and CLC, thereby forming a heterogeneous cryptographic environment [13], [14]. To enable secure and efficient data exchange across such heterogeneous slices, a range of heterogeneous signcryption schemes have been proposed. Based on the type of heterogeneity involved, we analyze the existing schemes in terms of computational efficiency, communication efficiency, and security. A comparative overview is presented in Table I.

To enable secure interactions between PKI and IBC in heterogeneous network slicing, Wang proposed a mutual heterogeneous signcryption scheme that satisfies most security properties and demonstrates strong security guarantees [15]. However, its reliance on bilinear pairings incurs significant computational overhead, making it less suitable for low-latency network slicings. To address this, Tao also proposed a mutual heterogeneous signcryption scheme that reduces the number of bilinear pairing operations to improve computational efficiency but fails to meet public verifiability and known session-specific temporary information security [16]. Building on this, Khan, Pan, and Ting proposed further optimizations that eliminate pairing operations altogether to improve efficiency [17]–[20]. However, their schemes exhibit security limitations, such as vulnerability to forgery and chosen-ciphertext attacks, which hinder their practical deployment [17]–[20].

For secure interaction between IBC and CLC in heterogeneous network slicing, Qiu proposed a heterogeneous signcryption scheme for multiple messages and multiple receivers based on Lagrange interpolation theorem, which eliminates the heavy computational burden brought by bilinear pairings and achieves high computational efficiency [21]. Nonetheless,

TABLE I: Comparison of Related Works in Efficient and Security.

Scheme	Type	Computational Efficiency		Communication Efficiency	Security
		Without Bilinear Pairing	Few Multiplications		
Wang [15]	PKI → IBC	×	✓	✓	✓
Tao [16]	PKI → IBC	×	✓	✓	✓
Khan [17]	IBC → PKI	✓	✓	✓	×
Pan [18]	IBC → PKI	✓	✓	✓	×
Ting [19]	IBC → PKI	✓	✓	✓	×
Qiu [21]	IBC → CLC	✓	✓	✓	×
Niu [24]	IBC → CLC	✓	×	✓	✓
Ullah [22]	CLC → IBC	✓	✓	×	×
Elkhalil [23]	CLC → IBC	✓	✓	×	×
Hou [25]	PKI → CLC	×	✓	✓	×
Liu [26]	PKI ↔ CLC	✓	×	✓	✓
Niu [28]	CLC → PKI	✓	×	✓	✓
Elkhalil [27]	CLC → PKI	✓	×	×	✓
MHSC	All	✓	✓	✓	✓

it lacks unforgeability, as malicious recipients can forge valid ciphertexts, posing risks in multi-tenant slicing environments. Similarly, Ullah and Elkhalil's scheme for vehicular networks also exhibits security vulnerabilities, restricting its applicability in network slicings [22], [23]. Niu proposed a hybrid group heterogeneous signcryption scheme that avoids bilinear pairings but relies on excessive multiplications, leading to computational inefficiencies [24].

For secure communication between PKI and CLC in heterogeneous network slicing, Hou proposed a heterogeneous signcryption scheme supporting equality testing for IoT [25]. However, its reliance on bilinear pairings results in high computational costs, making it impractical for latency-sensitive slicing environments. To improve efficiency, Liu introduced a scheme that eliminates bilinear pairings, relying only on group multiplications and additions [26]. Subsequent refinements by Elkhalil and Niu further enhanced computational efficiency and security, improving adaptability to heterogeneous slicing networks [27], [28]. However, their schemes still involve a considerable number of multiplications, which limits computational efficiency [27], [28].

Based on the analysis, it is evident that current schemes are still insufficient to support secure and efficient communication in multi-heterogeneous network slicings. Most approaches focus solely on data transmission between two distinct cryptographic systems, which limits interoperability across PKI, IBC, and CLC. Moreover, some schemes rely on computationally expensive bilinear pairings, rendering them impractical for latency-sensitive or resource-constrained network slicing. Further, certain schemes fail to ensure security properties, such as forward secrecy, thereby diminishing their applicability.

Unlike the aforementioned studies, we focus on constructing a generalized multi-heterogeneous signcryption mechanism that ensures secure and efficient data transmission across PKI, IBC, and CLC within heterogeneous network slicings.

III. Preliminaries

In this section, we provide a brief overview of the fundamental knowledge relevant to this paper, including the system model, security goals and security assumptions.

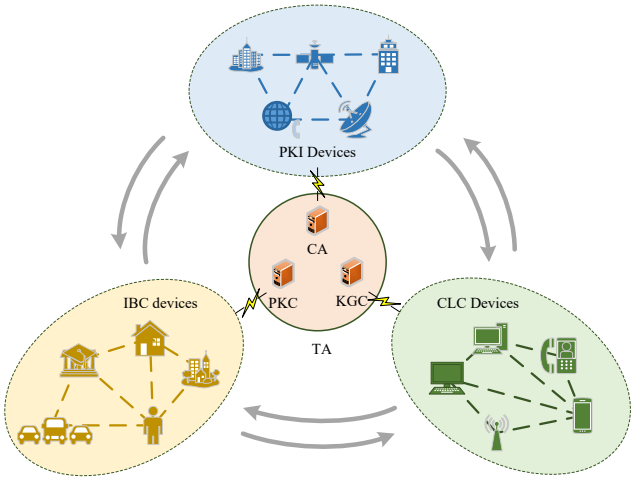


Fig. 1: The System Model for Multi-Heterogeneous Slicing Networks.

A. System Model

For network slicing environments, we give the system model for MHSC, which is based on three distinct public key cryptographic systems, as illustrated in Fig. 1. In the model, there are four key entities: Trusted Authority (TA), PKI devices, IBC devices, and CLC devices.

- *Trusted Authority:* TA manages cryptographic keys and network slicing. It consists of a Certification Authority (CA) in the PKI, a PKG in IBC, and a KGC in CLC.
- *PKI Devices:* Devices using PKI rely on certificates for secure communication, ideal for slices requiring high security and trust.
- *IBC Devices:* Devices in IBC use identities as public keys, simplifying key management, making it efficient for environments with frequent device changes.
- *CLC Devices:* Devices in CLC generate keys with the TA, balancing security and performance, suitable for slices needing flexibility and efficiency.

These heterogeneous devices collaborate to meet diverse security and performance requirements across network slicing.

B. Security Goals

To ensure the secure transmission of data across heterogeneous network slices, the design of MHSC's security goals include the following security properties.

1) *Confidentiality:* Only authorized users can decipher the ciphertext and obtain the correct message, while unauthorized users cannot access the correct information.
2) *Unforgeability:* Attackers cannot forge a legitimate ciphertext based on existing information.
3) *Non-Repudiation:* The involved entities cannot deny their participation in the message event they sent.
4) *Integrity:* The message transmitted over public channels is not illegally tampered with.
5) *Public Verifiability:* After obtaining a signcryption ciphertext, any user can verify its validity through certain computations.
6) *Forward Secrecy:* Even if the private key is compromised, attackers cannot access the message during previous communication.
7) *Known Session-Specific Temporary Information Security:* Even if the attacker knows the temporary random number used in signcryption process, attacker still cannot obtain the correct message.
8) *Scalability:* Maintains security and performance efficiency while accommodating an increasing number of heterogeneous slices and devices.

C. Security Assumptions

To ensure the security of MHSC, we give two security assumptions, which serve as the foundation for the subsequent security analysis. In these assumptions, we assume the existence of an additive cyclic group G of order q, with P as the generator of G. Additionally, the notations used throughout this paper are summarized in Table II.

- *Discrete Logarithm (DL) Assumptions:* Given $a \in \mathbb{Z}_q^*$, attacker A can efficiently compute aP. However, the probability Adv_A^{DL} of attacker A successfully computing the correct $a \in \mathbb{Z}_q^*$ from the tuple (P, aP) is negligible.
- *Computational Diffie-Hellman (CDH) Assumptions:* Given $a, b \in \mathbb{Z}_q^*$, attacker A can efficiently compute abP. However, the probability Adv_A^{CDH} of attacker A successfully computing the correct abP from the tuple (P, aP, bP) is also negligible.

IV. The Designed of MHSC

In this section, we propose the generalized multi-heterogeneous signcryption scheme, as MHSC. MHSC consists of four phases: *Set Up*, *User Registration*, *General Signcryption*, and *General Unsigncryption*. The overall process is illustrated in Fig. 2.

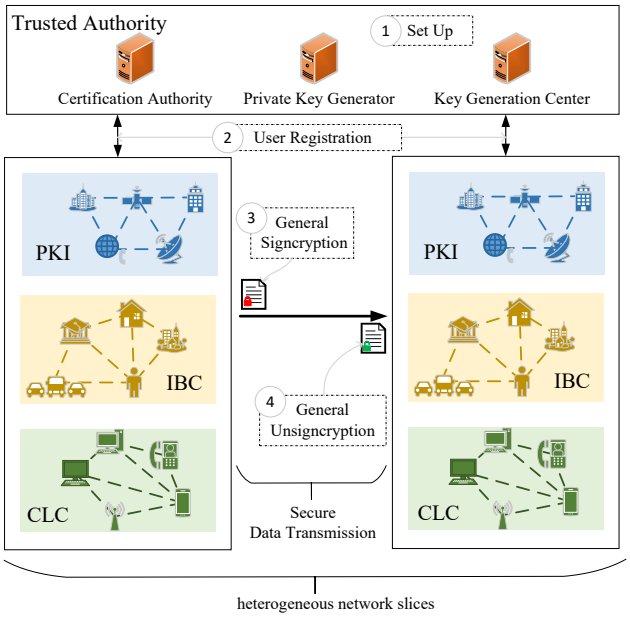


Fig. 2: The Overview of MHSC.

TABLE II: Notations and Explanations.

Symbol	Explanation
G	Additive cyclic group of order q
P	Generator of G
Z_q^*	The number of $[1, ...q-1]$
s	Master system key
P_{pub}	System public key
h_1, h_2, h_3, h_4, h_5	Hash functions
$F(\cdot)$	Generalized function
$params$	System public parameters
$\Phi(\cdot)$	Set of specific elements
σ	Signcryption ciphertext
m	Message that the device needs to send
ID_i	Identity of device in network slicing
(PK_S, SK_S)	Public and private keys of sender ID_S
(PK_R, SK_R)	Public and private keys of receiver ID_R
$\oplus$	XOR operation

A. Set Up

Based on the given system security parameter γ, TA selects an additive cyclic group G with a generator P, where the order of G is q ($q < 2^\gamma$). Next, TA selects five secure hash functions: $h_1 : \{0,1\}^l \times G \times G \to Z_q^*$, $h_2 : G \to \{0,1\}^*$, $h_3 : \{0,1\}^{l^2} \times G^4 \times \{0,1\}^* \times G \to Z_q^*$, $h_4 : \{0,1\}^{l^2} \times G^5 \times \{0,1\}^* \times G \to Z_q^*$, $h_5 : \{0,1\}^{l^2} \times G^6 \times \{0,1\}^* \times G \to Z_q^*$, where l represents the bit length of the unique device identifier.

Then, TA chooses a random number $s \in Z_q^*$ as the master system key, computes the system public key $P_{pub} = sP$ and define a generalized function $F(\cdot)$ as Equation 1. here, PK_i represents the public key of device ID_i, $\Phi(\cdot)$ denotes the set of specific elements. For example, $\Phi(G)$ represents the set of any one element from an additive cyclic group G.

$$F(PK_i) = \begin{cases} 1 & PK_i \in \Phi(G) \\ 2 & PK_i \in \Phi(G, G) \\ 3 & PK_i \in \Phi(G, G, G) \\ 0 & others. \end{cases} \tag{1}$$

Finally, TA generates the system public parameters $params = \{G, P, q, P_{pub}, h_1, h_2, h_3, h_4, h_5, F(\cdot)\}$ and distributes $params$ to all devices in network via secure channel.

B. User Registration

Considering the diversity of key generation across various heterogeneous network slices, MHSC divides the user registration into three parts: *PKI-Gen*, *IBC-Gen*, and *CLC-Gen*.

1) PKI-Gen: In PKI, device ID_i first selects a random number $x_i \in Z_q^*$ as the private key $SK_i = x_i$, and generates $P_i = {x_i}^{-1}P$ as the the corresponding public key. Afterwards, ID_i sends the generated public key $PK_i = P_i$ to the CA for signature and the generation of the corresponding public certificate.

2) IBC-Gen: In IBC, device ID_i uses the unique identifier ID_i as the private key request, which is sent to PKG. Upon receiving the private key request, the PKG selects a random number $r_i \in Z_q^*$ and computes $R_i = r_iP$. Then, PKG computes $h_i = h_1(ID_i, R_i, P_{pub})$, generates $d_i = r_i + sh_i$ and send (R_i, d_i) to ID_i through a secure channel. Upon receiving (R_i, d_i) sent by PKG, ID_i computes $X_i = h_i^{-1}R_i$ and generates the private key $SK_i = d_i$ and the public key $PK_i = (R_i, X_i)$.

3) CLC-Gen: In CLC, device ID_i selects a random number $x_i \in Z_q^*$ as secret value, generates the corresponding public key $P_i = x_iP$, and sends (ID_i, P_i) as the partial private key request to KGC. Upon receiving the partial private key request, KGC selects a random number $r_i \in Z_q^*$, computes $R_i = r_iP$ and $h_i = h_1(ID_i, R_i, P_i)$. Next, KGC generates the partial private key $d_i = r_i + sh_i$ and sends (d_i, R_i) to ID_i through a secure channel. Finally, ID_i computes $X_i = h_i^{-1}R_i$, combines the partial private key and secret value to generate the full private key $SK_i = (x_i, d_i)$ and the full public key $PK_i = (P_i, R_i, X_i)$.

After the user registration phase, based the target device's public key PK_i, any user can perform the following calculation to determine the cryptosystem and private key type used by the target device in network slicing.

- $F(PK_i) = 1$: device ID_i under PKI with public/private key $SK_i = x_i, PK_i = P_i$.
- $F(PK_i) = 2$: device ID_i under IBC with public/private key $SK_i = d_i, PK_i = (R_i, X_i)$.
- $F(PK_i) = 3$: device ID_i under CLC with public/private key $SK_i = (x_i, d_i), PK_i = (P_i, R_i, X_i)$.
- $F(PK_i) = 0$: device ID_i does not belong to any of the cryptosystems of PKI, IBC and CLC, and is not part of the discussion in this paper.

To better illustrate MHSC, we assume a multi-heterogeneous network slicing scenario involving two devices: the sender ID_S and the receiver ID_R. Both entities have completed *User Registration* and generated their respective public-private key pairs.

C. General Signcryption

In *General Signcryption*, the sender ID_S leverages its and the receiver's public keys PK_S, PK_R to determine heterogeneity and identify their respective cryptosystems (Steps 1–2). It then computes the signature and ciphertext using its private key SK_S and the receiver's public key PK_R, forming the signcryption ciphertext σ (Steps 3–6), as Fig. 3.

1) Selects a random number $a \in Z_q^*$, calculates $F(PK_S)$ and $F(PK_R)$.
2) Verifies whether the equation $F(PK_S) = F(PK_R)$ or $F(PK_S) * F(PK_R) = 0$ holds.
   - If holds, it means that sender ID_S and receiver ID_R are in the same public key cryptosystem, or there is one of the two that does not belong to the three public key cryptosystems PKI, IBC and CLC, and the process can be exited;
   - otherwise, ID_S performs the next step.
3) Generates the T_1 and T_2 as follows.
   a) If $F(PK_S) = 1$, calculates $T_1 = aP_S$ and T_2 as follows.
      - If $F(PK_R) = 2$, computes $h_R = h_1(ID_R, R_R, P_{pub})$ and $T_2 = ah_R x_S^{-1}(X_R + P_{pub})$.
      - If $F(PK_R) = 3$, computes $h_R = h_1(ID_R, R_R, P_R)$ and $T_2 = ah_R x_S^{-1}(h_R^{-1}P_R + X_R + P_{pub})$.

   b) If $F(PK_S) = 2$, calculates $h_S = h_1(ID_S, R_S, P_{pub})$, $T_1 = ah_S(X_S + P_{pub})$ and T_2 as Equation 2.

$$T_2 = \begin{cases} ad_S P_R & F(PK_R) = 1 \\ ah_R d_S(h_R^{-1}P_R + X_R + P_{pub}) & F(PK_R) = 3. \end{cases} \tag{2}$$

   Note that, $h_R = h_1(ID_R, R_R, P_R)$.

   c) If $F(PK_S) = 3$, calculates $T_1 = aP_S$ and T_2 as Equation 3.

$$T_2 = \begin{cases} ax_S P_R & F(PK_R) = 1 \\ ax_S h_R(X_R + P_{pub}) & F(PK_R) = 2. \end{cases} \tag{3}$$

   Note that, $h_R = h_1(ID_R, R_R, P_{pub})$.
4) Based the message m to be transmitted, generates the ciphertext $c = m \oplus h_2(T_2)$.
5) Calculates h according to Equation 5, and calculates signature as Equation 4.

$$S = \begin{cases} a + hx_S & F(PK_S) = 1 \\ d_S(a + h) & F(PK_S) = 2 \\ d_S + hax_S & F(PK_S) = 3. \end{cases} \tag{4}$$

6) Generates the ciphertext $\sigma = (c, S, T_1)$, and send σ to receiver ID_R through public channel.

D. General Unsigncryption

In *General Unsigncryption*, upon receiving $\sigma = (c, S, T_1)$, the receiver ID_R leverages its and the sender's public keys PK_S, PK_R via a generalized function $F(\cdot)$ to determine heterogeneity and identify the respective cryptographic systems (Step 1). Then, using its private key SK_R and the sender's public key PK_S, the receiver ID_R verifies the signature and recovers the message (Steps 2–5).

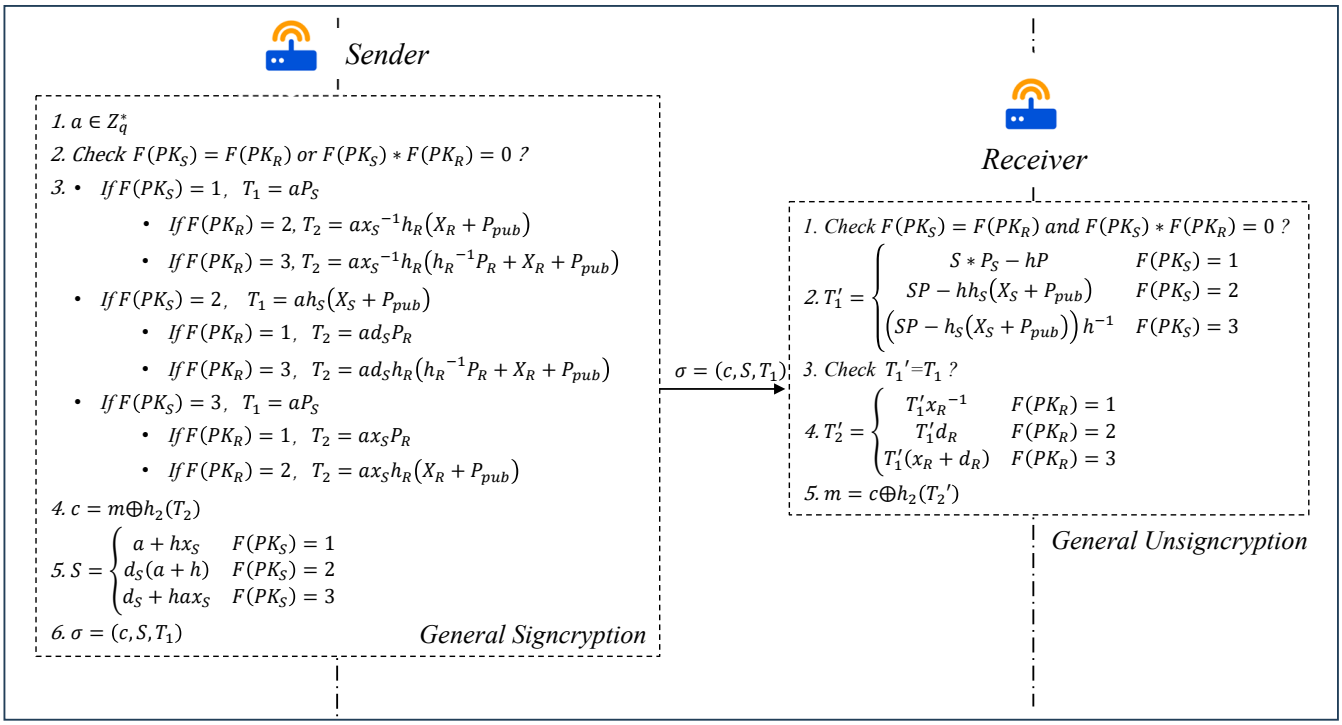


Fig. 3: The Main Flow of Signcryption and Unsigncryption of MHSC.

$$h = \begin{cases} h_3(ID_S, ID_R, P_S, R_R, X_R, P_{pub}, c, T_1) & F(PK_S) = 1, F(PK_R) = 2 \\ h_4(ID_S, ID_R, P_S, P_R, R_R, X_R, P_{pub}, c, T_1) & F(PK_S) = 1, F(PK_R) = 3 \\ h_3(ID_S, ID_R, R_S, X_S, P_R, P_{pub}, c, T_1) & F(PK_S) = 2, F(PK_R) = 1 \\ h_5(ID_S, ID_R, R_S, X_S, P_R, R_R, X_R, P_{pub}, c, T_1) & F(PK_S) = 2, F(PK_R) = 3 \\ h_4(ID_S, ID_R, P_S, R_S, X_S, P_R, P_{pub}, c, T_1) & F(PK_S) = 3, F(PK_R) = 1 \\ h_5(ID_S, ID_R, P_S, R_S, X_S, R_R, X_R, P_{pub}, c, T_1) & F(PK_S) = 3, F(PK_R) = 2. \end{cases} \tag{5}$$

1) Calculates $F(PK_S)$ and $F(PK_R)$ and verifies whether the equation $F(PK_S) = F(PK_R)$ or $F(PK_S) * F(PK_R) = 0$ holds.
   - If holds, it means that receiver and sender are in the same public key cryptosystem, or there is one of the two that does not belong to PKI, IBC and CLC, and the process can be exited;
   - otherwise, receiver performs the next step.
2) Calculates h based on the public keys PK_S and PK_R with Equation 5, and calculates T_1' as Equation 6.

$$T_1' = \begin{cases} SP_S - hP & F(PK_S) = 1 \\ SP - hh_S(X_S + P_{pub}) & F(PK_S) = 2 \\ (SP - h_S(X_S + P_{pub}))h^{-1} & F(PK_S) = 3. \end{cases} \tag{6}$$

   Note that,

$$h_S = \begin{cases} h_1(ID_S, R_S, P_{pub}) & F(PK_S) = 2 \\ h_1(ID_S, R_S, P_S) & F(PK_S) = 3. \end{cases} \tag{7}$$

3) Verifies if the equation $T_1' = T_1$ holds.
   - If false, the signcryption ciphertext σ is invalid, and the receiver discards it, outputting $\perp$;
   - otherwise, the receiver goes the next step.
4) Calculates T_2' as Equation 8.

$$T_2' = \begin{cases} T_1' \cdot x_R^{-1} & F(PK_R) = 1 \\ T_1' \cdot d_R & F(PK_R) = 2 \\ T_1' \cdot (x_R + d_R) & F(PK_R) = 3. \end{cases} \tag{8}$$

5) Recovers the message $m = c \oplus h_2(T_2')$.

E. Correctness Proofs

To verify the correctness of MHSC, we provide proofs for the correctness of significant equations, including the main computation processes for T_1 and T_2.

1) Correctness Verification of T_1: The specific computation process varies depending on the cryptosystem of the sender ID_S, i.e., $F(PK_S)$, as outlined below.

- When sender belongs to PKI, i.e., $F(PK_S) = 1$.

$$\begin{aligned} T_1' &= SP_S - hP \\ &= (a + hx_S) * x_S^{-1}P - hp = T_1. \end{aligned} \tag{9}$$

- When sender belongs to IBC, i.e., $F(PK_S) = 2$.

$$\begin{aligned} T_1' =& SP - hh_S(X_S + P_{pub}) \\ =& (a+h)d_S P - h(R_S + h_S P_{pub}) = T_1. \end{aligned} \tag{10}$$

- When sender belongs to CLC, i.e., $F(PK_S) = 3$.

$$\begin{aligned} T_1' =& (SP - h_S(X_S + P_{pub}))h^{-1} \\ =& ((d_S + ahx_S)P - (R_S + h_S P_{pub}))h^{-1} \\ =& ahx_S h^{-1} P = T_1. \end{aligned} \tag{11}$$

2) Correctness Verification of T_2: The specific computation process depends on the cryptosystem to which sender ID_S and receiver ID_R belong, i.e., the differences in $F(PK_S)$ and $F(PK_R)$, with different calculations performed as follows.

- When $F(PK_S) = 1$ and $F(PK_R) = 2$ (PKI $\rightarrow$ IBC).

$$\begin{aligned} T_2' =& d_R T_1' = d_R(SP_S - hP) \\ =& d_R a P_S = a x_S^{-1}(r_R + h_R s)P = T_2. \end{aligned} \tag{12}$$

- When $F(PK_S) = 1$ and $F(PK_R) = 3$ (PKI $\rightarrow$ CLC).

$$\begin{aligned} T_2' =& (x_R + d_R)T_1' = (x_R + d_R)(SP_S - hP) \\ =& (x_R + d_R)aP_S = T_2. \end{aligned} \tag{13}$$

- When $F(PK_S) = 2$ and $F(PK_R) = 1$ (IBC $\rightarrow$ PKI).

$$\begin{aligned} T_2' =& x_R^{-1} T_1' = x_R^{-1}(SP - hh_S(X_S + P_{pub})) \\ =& x_R^{-1} a h_S(X_S + P_{pub}) = T_2. \end{aligned} \tag{14}$$

- When $F(PK_S) = 2$ and $F(PK_R) = 3$ (IBC $\rightarrow$ CLC).

$$\begin{aligned} T_2' =& (x_R + d_R)T_1' \\ =& (x_R + d_R)(SP - hh_S(X_S + P_{pub})) \\ =& (x_R + d_R)ah_S(X_S + P_{pub}) = T_2. \end{aligned} \tag{15}$$

- When $F(PK_S) = 3$ and $F(PK_R) = 1$ (CLC $\rightarrow$ PKI).

$$\begin{aligned} T_2' =& T_1' x_R^{-1} \\ =& (SP - h_S(X_S + P_{pub}))h^{-1} x_R^{-1} \\ =& x_R^{-1} a P_S = a x_S P_R = T_2. \end{aligned} \tag{16}$$

- When $F(PK_S) = 3$ and $F(PK_R) = 2$ (CLC $\rightarrow$ IBC).

$$\begin{aligned} T_2' =& T_1' d_R = (SP - h_S(X_S + P_{pub}))h^{-1} d_R \\ =& a P_S d_R = a x_S (r_R + h_R s)P = T_2. \end{aligned} \tag{17}$$

According to the above calculation proofs, MHSC is computationally correct and valid.

V. Security Analysis

In this section, to enhance the simplicity and readability of the security analysis, we focus on the specific signcryption process $MHSC_{PKI \rightarrow IBC}$ from PKI to IBC (i.e., when $F(PK) = 1$ and $F(PK) = 2$) as the case study for MHSC. Moreover, we provide security proofs and discussions for $MHSC_{PKI \rightarrow IBC}$.

A. Security Proofs

Based on ROM, we provide the formal proofs of confidentiality and unforgeability of $MHSC_{PKI \rightarrow IBC}$, the details as follows [29].

1) Confidentiality: Based on ROM, we construct the game $Game^{IND-CCA-2}_{PKI \rightarrow IBC}$ between the adversary A and the challenger C and present the following theorem to prove the confidentiality of $MHSC_{PKI \rightarrow IBC}$. Due to space limitation, we only give the Theorem 1, and please refer to Appendix A for the detailed proof.

Theorem 1: In ROM, assuming an attacker A who can perform $q_i (i = 1, 2, 3)$ hash queries, q_{sk}^{PKI} PKI private key queries, q_{pk}^{PKI} PKI public key queries, q_e^{IBC} IBC key extraction queries, q_{pk}^{IBC} IBC public key queries, q_{sc} signcryption queries, and q_{usc} unsigncryption queries, and can break the $Game^{IND-CCA-2}_{PKI \rightarrow IBC}$ with a non-negligible advantage ε, then there exists a challenger C who can solve the CDH problem with the following advantage:

$$\varepsilon' = \frac{\varepsilon}{q_1 q_2}(1 - \frac{1}{q_1})^{q_e^{IBC}}(1 - \frac{q_{sc}(q_2 + q_3)}{2^\gamma}). \tag{18}$$

However, CDH is a hard security assumption, and thus, attacker A can not break the $Game^{IND-CCA-2}_{PKI \rightarrow IBC}$. In other words, $MHSC_{PKI \rightarrow IBC}$ satisfies confidentiality.

2) Unforgeability: Similarly, we prove the unforgeability of $MHSC_{PKI \rightarrow IBC}$ by constructing a game $Game^{EUF-CMA}_{PKI \rightarrow IBC}$ between the adversary A and the challenger C and present the theorem 2 to prove the unforgeability. here, we only briefly describe the Theorem 2 and please refer to Appendix B for the detailed proof.

Theorem 2: In ROM, assuming an attacker A that can perform $q_i (i = 1, 2, 3)$ hash queries, q_{sk}^{PKI} PKI private key queries, q_{pk}^{PKI} PKI public key queries, q_e^{IBC} IBC key extraction queries, q_{pk}^{IBC} IBC public key queries, q_{sc} signcryption queries, and q_{usc} unsigncryption queries, with a non-negligible advantage ε to break the $Game^{EUF-CMA}_{PKI \rightarrow IBC}$, then there exists a challenger C that can solve the DL problem with the following advantage:

$$\varepsilon' = \varepsilon * (1 - \frac{1}{q_{sk}^{PKI}})^2 * (1 - q_{sc} * \frac{(q_2 + q_3)}{2^\gamma}). \tag{19}$$

However, the DL is a hard security assumption, so attacker A cannot break the $Game^{EUF-CMA}_{PKI \rightarrow IBC}$. Thus, $MHSC_{PKI \rightarrow IBC}$ satisfies unforgeability.

B. Security Discussions

1) Non-Repudiation: To prevent devices from denying their involvement in generating signcryption ciphertext, non-repudiation is essential. In MHSC, the ciphertext $\sigma = (c, S, T_1)$ consists of c and T_1, derived from the message and a random number, ensuring randomness and preventing direct traceability to a specific user. However, the signature S enables verification of the sender's identity. In the PKI-to-IBC transmission, S is computed as $S = a + hx_S$, where a is a random value and x_S is the sender's private key. The security of MHSC against forgery relies on the hardness of the DL problem—an attacker without knowledge of x_S and a

cannot generate a valid signature S to pass verification. Only the sender possessing x_S and a can compute a legitimate S, ensuring that the generated signcryption ciphertext cannot be denied. Thus, MHSC guarantees non-repudiation by leveraging the DL assumption, preventing the sender from denying having sent the message.

2) Integrity: In MHSC, to preserve the integrity of message m during transmission, three hash functions (h_3, h_4, h_5) are integrated into Equation 5. Additionally, the sender embeds h into the signature S, ensuring that any modification to the plaintext invalidates the signcryption ciphertext due to its unforgeability. Consequently, an attacker cannot alter the ciphertext while generating a valid signature. Verification of the equation detects any tampering, thereby ensuring data integrity during transmission.

3) Public Verifiability: To prevent illegal data from being transmitted over an open channel, public verifiability ensures that any user can perform certain computations to verify the validity of the ciphertext and exclude illegal data. In MHSC, upon receiving the signcryption ciphertext $\sigma = (c, S, T_1)$, any user first computes T_1' according to Equation 20 and verifies $T_1 = T_1'$ to determine the validity of the ciphertext.

$$T_1' = \begin{cases} S \cdot P_S - hP & F(PK_S) = 1 \\ SP - hh_S(X_S + P_{pub}) & F(PK_S) = 2 \\ (SP - h_S(X_S + P_{pub}))h^{-1} & F(PK_S) = 3. \end{cases} \quad (20)$$

There, MHSC can meet the public verifiability, and realize the identification and exclusion of illegal data in heterogeneous network slicing.

4) Forward Secrecy: In MHSC, forward secrecy is ensured through the following proof. Taking $MHSC_{PKI \to IBC}$ as an example, message encryption and decryption are represented as $c = m \oplus h_2(T_2)$ and $m = c \oplus h_2(T_2')$, respectively. This implies that an attacker A must compute the correct T_2 to decrypt the message.

However, T_2 is computed as $T_2 = d_R T_1 = a{x_S}^{-1}(r_R + sh_R)P$. Due to the hardness of the CDH, even if the sender's private key x_S is compromised, the attacker A cannot derive the correct T_2 without knowing the random value a or the private key $SK_R = r_R + sh_R$. Consequently, the attacker cannot recover the original message $m = c \oplus h_2(T_2')$. Furthermore, in MHSC, the generation of T_2 follows Equation 8. As a result, based on the CDH, an attacker A remains incapable of computing the correct T_2 and, therefore, cannot retrieve any previously transmitted messages. Thus, MHSC guarantees forward secrecy.

5) Known Session-Specific Temporary Information Security: In heterogeneous network slices, various devices coexist, some of which—such as wearable devices—are particularly susceptible to side-channel attacks. Attackers may exploit these vulnerabilities to extract random values used in the encryption process, potentially leading to unauthorized access to sensitive private information. Similar to the discussion on forward secrecy, an attacker's ability to recover the original message from the ciphertext $c = m \oplus h_2(T_2)$ hinges on whether they can compute the correct T_2. Although an attacker A may obtain the random value a and the public keys of both the sender and receiver, they lack the sender's or receiver's private keys SK_S and SK_R. Due to the hardness of the CDH, A cannot compute the correct T_2 and, consequently, cannot recover the original message m. Thus, MHSC satisfies known session-specific temporary information security.

6) Scalability: In MHSC, the trusted authority consists of CA, PKG, and KGC, corresponding to PKI, IBC, and CLC systems, respectively. Each entity—CA, PKG, and KGC—independently manages keys within its designated network slicing. When a device joins or leaves the network, it interacts only with its respective key management entity following the regular key management process to update its keys, without affecting the independent operation of other slices. Furthermore, MHSC focuses on secure data interactions between heterogeneous slices while ensuring that internal communications within each slice remain unaffected. Even in large-scale device communication scenarios, MHSC does not require additional data storage. As a result, MHSC ensures efficient key management and scalability in large-scale network slicing environments, demonstrating its practical applicability.

VI. Performance Evaluation

In this section, to evaluate the performance of MHSC, we conduct comparative analysis in security properties, communication efficiency, and computational efficiency through theoretical comparison and simulation experiments [15]–[19], [21], [22], [24]–[28], [30]–[35].

A. Security Properties Comparison

Security strength plays a crucial role in determining the suitability of heterogeneous signcryption. To this end, we categorize MHSC into six type based on the data interactions between PKI, IBC and CLC. Then, by collating and analyzing the security properties, we compare the security properties of MHSC with recent schemes of the different type, and generate Table III [15]–[19], [21], [22], [24]–[28], [30]–[35].

From Table III, it can be observed that MHSC satisfies confidentiality, unforgeability, non-repudiation, integrity, public verifiability, forward secrecy, known session-specific temporary information security (KSSTIS) and scalability. However, for the compared schemes, most of them satisfy confidentiality, unforgeability, non-repudiation, integrity, and forward secrecy, and have certain deficiencies in terms of public verifiability and known session-specific temporary information security [16]–[19], [22], [24]–[28], [30]–[35]. Therefore, MHSC exhibits more security properties compared to other schemes in the respective types above, and can meet the requirements of secure data transmission in heterogeneous network slicings.

B. Comparison of Communication Efficiency

To evaluate the computational efficiency of heterogeneous signcryption in network slicing environments, the size of the ciphertext transmitted through the public channel becomes a critical factor to consider. Therefore, we assess the communication efficiency of various schemes by comparing the sizes

TABLE III: Security Properties Comparison for Different Heterogeneous Signcryption Scheme.

Type	Scheme	Confidentiality	Unforgeability	Non-Rep	Integrity	Pub-Verif	Forward Secrecy	KSSTIS	Scalability
PKI → IBC	Wang [15]	✓	✓	✓	✓	×	✓	✓	✓
	Tao [16]	✓	✓	✓	✓	×	×	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓
PKI → CLC	Hou [25]	✓	✓	✓	✓	×	✓	×	✓
	Liu [26]	✓	✓	✓	✓	×	✓	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓
IBC → PKI	Khan [17]	✓	✓	✓	✓	×	✓	×	✓
	Pan [18]	✓	✓	✓	✓	×	✓	×	✓
	Ting [19]	✓	✓	✓	✓	×	✓	×	✓
	Aithekar [30]	✓	✓	✓	✓	×	✓	×	✓
	Eltayieb [31]	✓	✓	✓	✓	×	✓	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓
IBC → CLC	Qiu [21]	✓	✓	✓	✓	✓	✓	×	✓
	Niu [24]	✓	✓	✓	✓	×	✓	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓
CLC → PKI	Elkhalil [27]	✓	✓	✓	✓	×	✓	×	✓
	Khalafalla [32]	✓	✓	✓	✓	×	×	×	✓
	Ali [33]	✓	✓	✓	✓	×	✓	×	✓
	Niu [28]	✓	✓	✓	✓	×	✓	×	✓
	Liu [26]	✓	✓	✓	✓	×	✓	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓
CLC → IBC	Ullah [22]	✓	✓	✓	✓	×	✓	×	✓
	Wang [34]	✓	✓	✓	✓	×	✓	×	✓
	Niu [35]	✓	✓	✓	✓	×	✓	×	✓
	MHSC	✓	✓	✓	✓	✓	✓	✓	✓

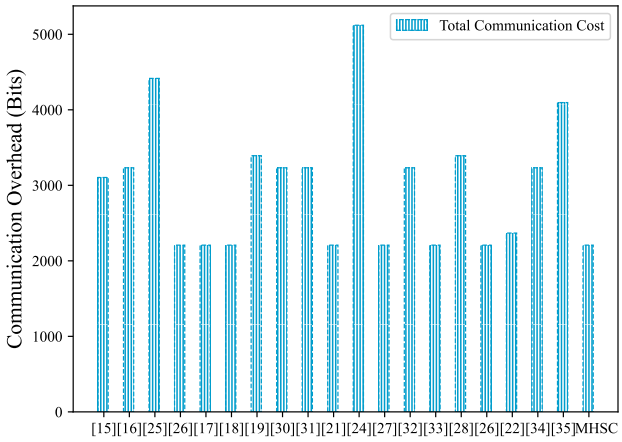


Fig. 4: Comparison of the Total Communication Overhead.

of transmitted elements and provide a theoretical comparison of total communication cost, as shown in Table IV [15]–[19], [21], [22], [24]–[28], [30]–[35]. here, $|G|$ denotes the length of elements in G, $|q|$ refers to the size of elements in Z_q^*, $|m|$ represents the size of message transmitted by devices, and $|ID|$ signifies the size of the device's unique identifier.

To conduct a more precise quantitative analysis, we used the element lengths in the 80-bit security model ($qbits = 512, rbits = 160$) to generate a comparison of the total communication length, as shown in Fig. 4. In this security model, the lengths of $|G|$ and $|q|$ are 1024 bits and 160 bits, respectively. Additionally, to eliminate communication discrepancies caused by varying message and identifier lengths, we assume message length $|m|$ of 1024 bits and device identifier length $|ID_i|$ of 32 bits. From Table IV and Fig. 4, it can be observed that the length of MHSC is only 2208 bits during communication. Compared with other schemes, MHSC has the smallest length of ciphertext data transmitted in communication process, and can improve the communication efficiency by 5.94% on average among the six types.

C. Comparison of Computational Efficiency

Computational efficiency is the crucial factor when evaluating the feasibility and applicability of heterogeneous signcryption schemes within network slices. Thus, we present the comparative analysis of the computational efficiency of MHSC through theoretical analysis and simulation experiments.

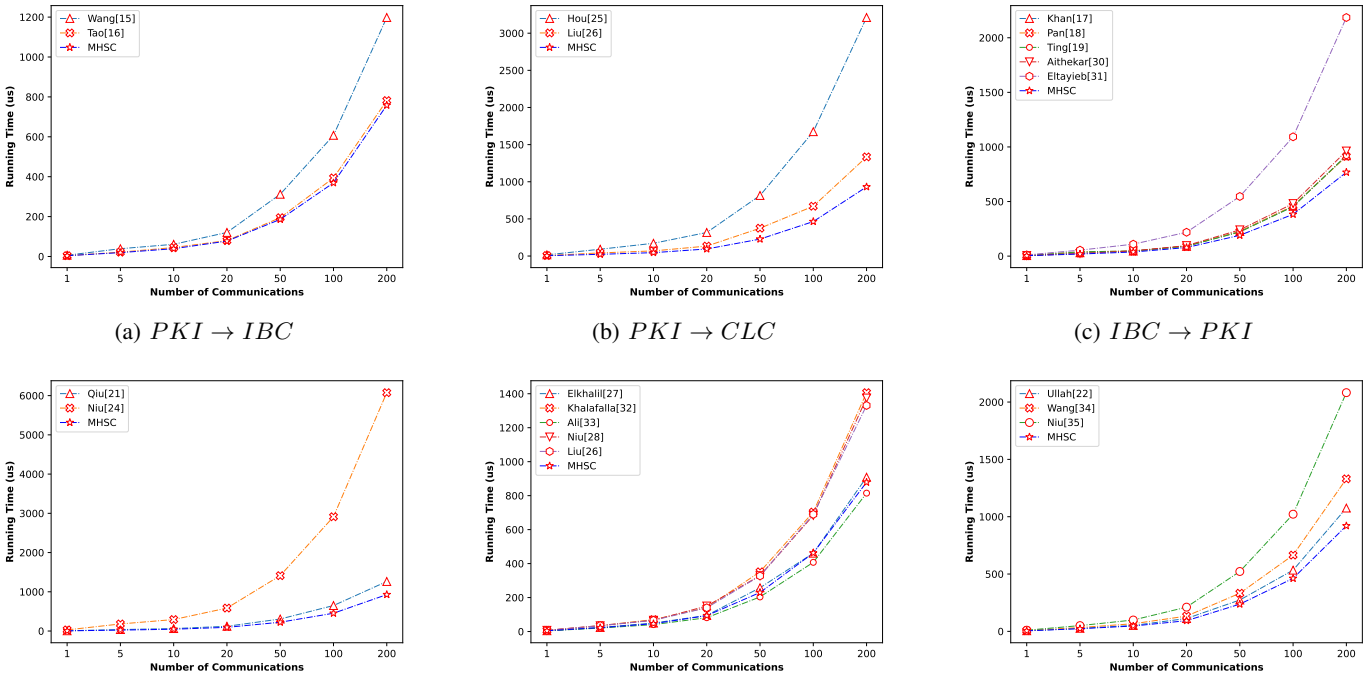
To facilitate the theoretical analysis of the computational efficiency of the signcryption scheme, most existing approaches primarily focus on operations within cyclic groups. These operations include addition, multiplication, exponentiation, and bilinear pairing. Therefore, in theoretical analysis, we conduct a statistical evaluation of the number of executions for these operations in both MHSC and several recently schemes, generating a theoretical runtime comparison Table IV [15]–[19], [21], [22], [24]–[28], [30]–[35]. here, A, M, E, P represent addition, multiplication, exponentiation, and bilinear pairing operation in cyclic group, respectively.

It can be seen from Table IV that MHSC is limited to addition and multiplication in the group. Notably, Ali's scheme preprocesses device public keys in CLC, eliminating the need for the system public key during signcryption and unsigncryption, thereby reducing computation to just 5 multiplications, fewer than MHSC's 6 multiplications [33]. Compared to existing schemes, except for the Ali's scheme in CLC-to-PKI transmission type, MHSC achieves higher computational efficiency in other transmission types by requiring fewer point multiplication and addition operations in theoretical analysis.

In the simulation experiment, we construct a virtual execution environment using physical machines (IOS: Windows 11, CPU: i5-12400, RAM: 32G, ROM: 2T), VMware Workstation 16 Pro, virtual machines (IOS: Ubuntu 20.04, Python 3.8,

TABLE IV: Comparison of Theoretical Computation and Communication Efficiency.

Type	Scheme	Computation Efficiency			Communication Efficiency	
		Signcryption	UnSigncryption	Computation Overhead	Data Type	Communication Overhead
PKI → IBC	Wang [15]	$3M+E+P$	$A+M+2P$	$A+4M+E+4P$	$\|m+2G+ID\|$	3104 bits
	Tao [16]	$P+E+M$	$A+3M+P$	$A+4M+E+2P$	$\|2m+G+q\|$	3232 bits
	MHSC	$A+2M$	$A+3M$	$2A+5M$	$\|m+G+q\|$	2208 bits
PKI → CLC	Hou [25]	$5M+2E+2P$	$A+5M+2P$	$A+10M+2E+4P$	$\|m+3G+2q\|$	4416 bits
	Liu [26]	$2A+4M$	$3A+5M$	$5A+9M$	$\|m+G+q\|$	2208 bits
	MHSC	$2A+3M$	$A+3M$	$3A+6M$	$\|m+G+q\|$	2208 bits
IBC → PKI	Khan [17]	$2M$	$3A+4M$	$3A+6M$	$\|m+G+q\|$	2208 bits
	Pan [18]	$2M$	$2A+4M$	$2A+6M$	$\|m+G+q\|$	2208 bits
	Ting [19]	$2M$	$2A+4M$	$2A+6M$	$\|m+2G+2q\|$	3392 bits
	Aithekar [30]	$2M$	$2A+4M$	$2A+6M$	$\|m+2G+2q\|$	3232 bits
	Eltayieb [31]	$2M+E$	$3M+E+2P$	$5M+2E+2P$	$\|m+2G+2q\|$	3232 bits
	MHSC	$A+2M$	$2A+3M$	$3A+5M$	$\|m+G+q\|$	2208 bits
IBC → CLC	Qiu [21]	$A+3M$	$2A+4M$	$3A+7M$	$\|m+G+q\|$	2208 bits
	Niu [24]	$9M$	$A+8M$	$A+17M$	$\|m+4G\|$	5120 bits
	MHSC	$3A+3M$	$2A+3M$	$5A+6M$	$\|m+G+q\|$	2208 bits
CLC → PKI	Elkhalil [27]	$2M$	$2A+4M$	$2A+6M$	$\|m+G+q\|$	2208 bits
	Khalafalla [32]	$A+4M$	$A+5M$	$2A+9M$	$\|m+2G+2q\|$	3232 bits
	Ali [33]	$2M$	$2A+3M$	$2A+5M$	$\|m+2G+2q\|$	2208 bits
	Niu [28]	$4M$	$5A+5M$	$5A+9M$	$\|m+2G+2q\|$	3392 bits
	Liu [26]	$3M$	$6A+6M$	$6A+9M$	$\|m+G+q\|$	2208 bits
	MHSC	$2M$	$2A+4M$	$2A+6M$	$\|m+G+q\|$	2208 bits
CLC → IBC	Ullah [22]	$A+3M$	$2A+3M$	$3A+6M$	$\|m+G+2q\|$	2368 bits
	Wang [34]	$3M+P$	$3M+P$	$6M+2P$	$\|m+3G\|$	3232 bits
	Niu [35]	$3M+E+P$	$A+M+3P$	$A+4M+E+4P$	$\|m+3G\|$	4096 bits
	MHSC	$A+2M$	$2A+4M$	$3A+6M$	$\|m+G+q\|$	2208 bits



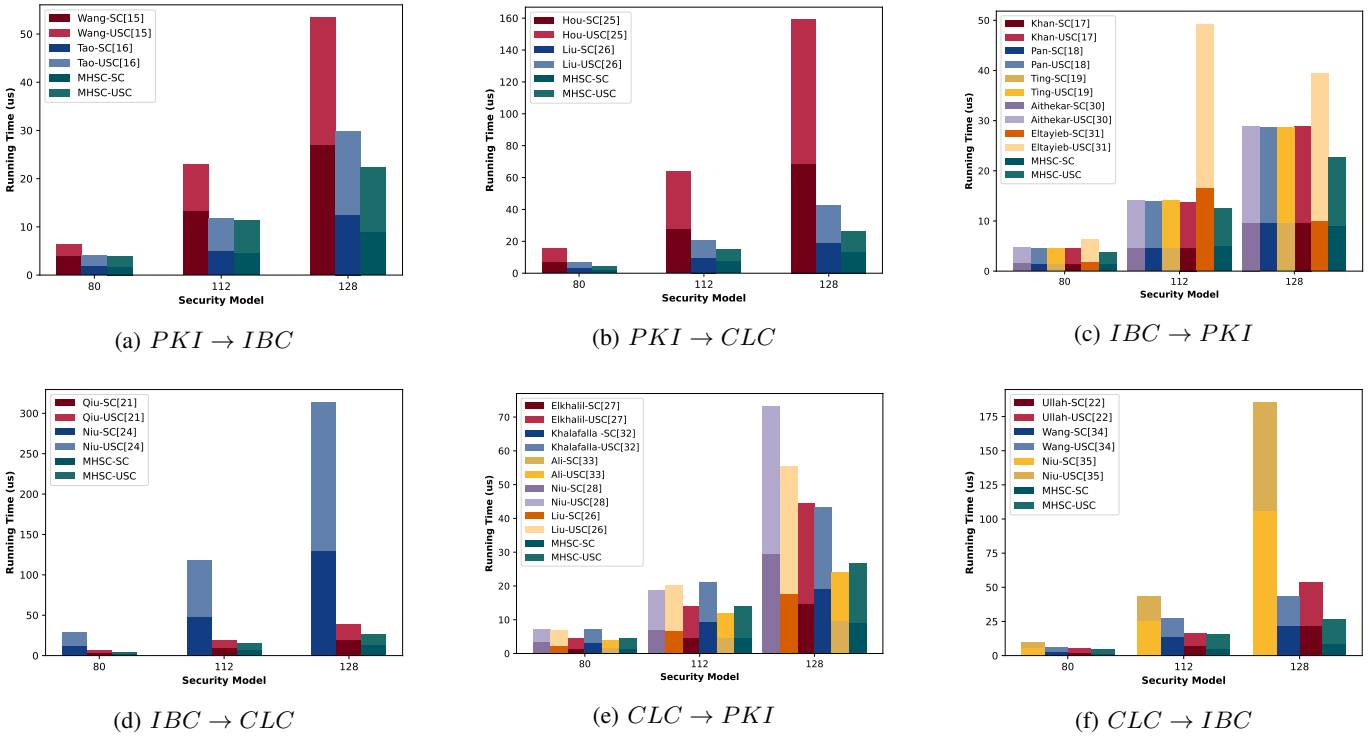
(a) $PKI \rightarrow IBC$ (b) $PKI \rightarrow CLC$ (c) $IBC \rightarrow PKI$

(d) $IBC \rightarrow CLC$ (e) $CLC \rightarrow PKI$ (f) $CLC \rightarrow IBC$

Fig. 5: Comparison of Running Time under Different Communication Counts

RAM 8GB, ROM: 40G), and the Pypbc 0.2 library. Within this environment, we implement the signcryption and unsigncryption processes for MHSC and the comparison schemes in PyCharm 2024.3.5. To evaluate computational efficiency under different security strengths, we established security models for 80-bit security ($qbits = 512, rbits = 160$), 112-bit security ($qbits = 1024, rbits = 224$), and 128-bit security ($qbits = 1536, rbits = 256$). Finally, assuming communication transmission time is negligible, we measure and compare the average running time of MHSC and other schemes under different execution counts and security models [15]–[19], [21], [22], [24]–[28], [30]–[35].

To evaluate computational efficiency across different execution iterations, we conduct experiments based on the 80-bit security model ($qbits = 512, rbits = 160$). Using a consistent execution process across all schemes, we measure and compare the average execution time over 1,000 iterations for MHSC and the other schemes under varying communica-



(a) $PKI \rightarrow IBC$ (b) $PKI \rightarrow CLC$ (c) $IBC \rightarrow PKI$

(d) $IBC \rightarrow CLC$ (e) $CLC \rightarrow PKI$ (f) $CLC \rightarrow IBC$

Fig. 6: Comparison of Running Time under Different Security Models.

tion counts, as shown in Fig. 5. The results indicate that as the number of communication counts increases, the running time for all schemes exhibits an increasing trend. While Ali's scheme demonstrates a shorter running time for CLC-to-PKI transmission, MHSC achieves the shortest running time in other transmission types, demonstrating superior overall computational efficiency, making it the more efficient choice in multi-heterogeneous environments [33].

Furthermore, to evaluate the computational efficiency of different schemes under varying security levels, we implement a main function to sequentially execute each scheme within security models of 80-bit ($qbits = 512, rbits = 160$), 112-bit ($qbits = 1024, rbits = 224$), and 128-bit ($qbits = 1536, rbits = 256$). We then analyse and compare the average running time over 1,000 iterations, as shown in Fig. 6. The results indicate that, similar to Fig. 5, MHSC achieves the shortest running time across different security models, except for Ali's scheme in CLC-to-PKI transmission. Overall, in six different transmission types, MHSC demonstrates an average computational efficiency improvement of 17.66% compared with the current scheme with the shortest running time [33].

VII. Conclusion

With the deployment of 5G technology and the rapid advancement of 6G networks, network slicing has become a key technology for addressing the diverse and evolving demands, exhibiting a trend toward heterogeneous development. In this context, ensuring the secure and efficient transmission of data between heterogeneous network slices is crucial. To this end, we propose a generalized multi-heterogeneous signcryption scheme, as MHSC, which can achieve secure and efficient data transmission between heterogeneous devices across any combination of PKI, IBC, and CLC. In terms of security, MHSC satisfies confidentiality, unforgeability, non-repudiation, integrity, forward security, public verifiability, and known session-specific temporary information security, thus meeting most of the security requirements for slicing networks. Through theoretical analysis and simulation experiments, MHSC demonstrates higher computational efficiency and communication efficiency compared to recently proposed schemes, making it a more scalable and resource-efficient solution for heterogeneous network slices. In future work, we plan to enhance the MHSC by incorporating features such as multi-message and multi-receiver capabilities, as well as improved anonymity, to address the evolving and complex requirements of network slicing environments.

Acknowledgments

This work was supported by the National Key R&D Program of China (2024YFB3108202). The authors also extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number RGP. 2/637/46

References

[1] G. Sun, Z. Xu, H. Yu, and V. Chang, "Dynamic network function provisioning to enable network in box for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7155–7164, 2020.

[2] S. Islam, Z. A. Atallah, A. K. Budati, M. K. Hasan, R. Kolandaisamy, and S. Nurhizam, "Mobile networks toward 5g/6g: Network architecture, opportunities and challenges in smart city," *IEEE Open Journal of the Communications Society*, 2024.

[3] W. Rafique, J. Barai, A. O. Fapojuwo, and D. Krishnamurthy, "A survey on beyond 5g network slicing for smart cities applications," *IEEE Communications Surveys & Tutorials*, 2024.

[4] A. Arbaoui, S. Abdelatif, and M. Derdour, "Network slicing solutions for internet of vehicles (iov) networks: A review," in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2024, pp. 1–7.

[5] W. Xia, L. Pu, X. Zou, P. Shilane, S. Li, H. Zhang, and X. Wang, "The design of fast and lightweight resemblance detection for efficient post-deduplication delta compression," *ACM Transactions on Storage*, vol. 19, no. 3, pp. 1–30, 2023.

[6] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, "Security and privacy in device-to-device (d2d) communication: A review," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1054–1079, 2017.

[7] X. Liu, P. Liu, B. Yang, and Y. Chen, "One multi-receiver certificateless searchable public key encryption scheme for iomt assisted by llm," *Journal of Information Security and Applications*, vol. 90, p. 104011, 2025.

[8] C. Paar, J. Pelzl, and T. Güneysu, "Introduction to public-key cryptography," in *Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms*. Springer, 2024, pp. 177–203.

[9] V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, "Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28 177–28 193, 2021.

[10] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: Principle, practice and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5215–5261, 2022.

[11] C. De Alwis, P. Porambage, K. Dev, T. R. Gadekallu, and M. Liyanage, "A survey on network slicing security: Attacks, challenges, solutions and research directions," *IEEE Communications Surveys & Tutorials*, 2023.

[12] B. Gong, C. Guo, C. Guo, C. Guo, Y. Sun, M. Waqas, and S. Chen, "Slim: A secure and lightweight multi-authority attribute-based signcryption scheme for iot," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1299–1312, 2024.

[13] M. Dai, G. Sun, H. Yu, and D. Niyato, "Maximize the long-term average revenue of network slice provider via admission control among heterogeneous slices," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 745–760, 2023.

[14] M. Dai, L. Luo, J. Ren, H. Yu, and G. Sun, "Psaccf: Prioritized online slice admission control considering fairness in 5g/b5g networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 6, pp. 4101–4114, 2022.

[15] C. Wang, C. Liu, S. Niu, L. Chen, and X. Wang, "An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 723–728.

[16] F. Tao, T. Shi, and S. Li, "Provably secure cross-domain authentication key agreement protocol based on heterogeneous signcryption scheme," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1. IEEE, 2020, pp. 2261–2266.

[17] M. A. Khan, I. Ullah, A. M. Abdullah, S. A. H. Mohsan, and F. Noor, "An efficient and conditional privacy-preserving heterogeneous signcryption scheme for the internet of drones," *Sensors*, vol. 23, no. 3, p. 1063, 2023.

[18] X. Pan, Y. Jin, Z. Wang, and F. Li, "A pairing-free heterogeneous signcryption scheme for unmanned aerial vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 426–19 437, 2022.

[19] P.-Y. Ting, J.-L. Tsai, and T.-S. Wu, "Signcryption method suitable for low-power iot devices in a wireless sensor network," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2385–2394, 2017.

[20] S. Tu, A. Badshah, H. Alasmary, and M. Waqas, "Eake-wc: Efficient and anonymous authenticated key exchange scheme for wearable computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4752–4763, 2024.

[21] J. Qiu, K. Fan, K. Zhang, Q. Pan, H. Li, and Y. Yang, "An efficient multi-message and multi-receiver signcryption scheme for heterogeneous smart mobile iot," *IEEE Access*, vol. 7, pp. 180 205–180 217, 2019.

[22] I. Ullah, M. A. Khan, N. Kumar, A. M. Abdullah, A. A. AlSanad, and F. Noor, "A conditional privacy preserving heterogeneous signcryption scheme for internet of vehicles," *IEEE Transactions on Vehicular Technology*, 2022.

[23] A. Elkhalil, R. Elhabob, N. Eltayieb *et al.*, "An efficient signcryption of heterogeneous systems for internet of vehicles," *Journal of Systems Architecture*, vol. 113, p. 101885, 2021.

[24] S. Niu, X. Yang, C. Wang, M. Tian, and X. Du, "Hybrid group signcryption scheme based on heterogeneous cryptosystem," *Journal of Electronics & Information Technology*, vol. 41, no. 5, pp. 1180–1186, 2019.

[25] Y. Hou, X. Huang, Y. Chen, S. Kumar, and H. Xiong, "Heterogeneous signcryption scheme supporting equality test from pki to clc toward iot," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 8, p. e4190, 2021.

[26] J. Liu, L. Zhang, R. Sun, X. Du, and M. Guizani, "Mutual heterogeneous signcryption schemes for 5g network slicings," *IEEE Access*, vol. 6, pp. 7854–7863, 2018.

[27] A. Elkhalil and J. Zhang, "Practical heterogeneous signcryption system for vehicular communication in vanets," *Computing*, vol. 105, no. 1, pp. 89–113, 2023.

[28] S. Niu, H. Shao, Y. Hu, S. Zhou, and C. Wang, "Privacy-preserving mutual heterogeneous signcryption schemes based on 5g network slicing," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 086–19 100, 2022.

[29] G. Bleumer, "Random oracle model," in *Encyclopedia of Cryptography, Security and Privacy*. Springer, 2023, pp. 1–2.

[30] A. Aithekar, P. Gupta, and D. Chaudhary, "A secure and efficient heterogeneous id-based signcryption for unmanned aerial vehicular networking system," *Security and Privacy*, vol. 7, no. 5, p. e389, 2024.

[31] N. Eltayieb, R. Elhabob, Y. Liao, F. Li, and S. Zhou, "A heterogeneous signcryption scheme with cryptographic reverse firewalls for iot and its application," *Journal of Information Security and Applications*, vol. 83, p. 103763, 2024.

[32] W. Khalafalla, W.-X. Zhu, A. Elkhalil, and I. Elfadul, "Efficient access control scheme for heterogeneous signcryption based on blockchain in vanets," *Cluster Computing*, pp. 1–21, 2024.

[33] I. Ali, Y. Chen, C. Pan, and S. Chen, "Cost-effective and secure scheme for fog computing-enabled internet of vehicles using clc-to-pki-based heterogeneous signcryption," *IEEE Transactions on Intelligent Vehicles*, 2024.

[34] Y. Wang, X. Jia, Y. Bao, Y. Cao, and J. Wen, "Efficient and provably secure offline/online heterogeneous signcryption scheme for vanets," *IEEE Internet of Things Journal*, 2024.

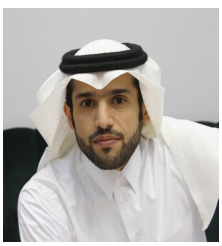
[35] S. Niu, Z. Li, M. Tian, C. Wang, and X. Jia, "An efficient heterogeneous signcryption scheme from certificateless to identity-based cryptosystem," in *MATEC Web of Conferences*, vol. 139. EDP Sciences, 2017, p. 00037.

Bei Gong received the BS degree from Shandong University, Qingdao, China, in 2005, and the PhD degree from the Beijing University of Technology, Beijing, China, in 2012. He has six National invention patents and one monograph textbook. He is the principal investigator of eight national projects such as the National Natural Science Foundation grants and six provincial and ministerial projects such as the General Science and Technology Program of Beijing Municipal Education Commission. Over the past five years, he has authored or coauthored more than 30 articles in top-tier journals and prestigious conferences in relevant research fields. His research interests include trusted computing, Internet of Things security, mobile Internet of Things, and mobile edge computing.

Yong Wu recieved his Master's degree from Beijing University of Technology, Beijing, China. He is currently pursuing his PhD degree in the field of trusted computing, systems and network security.

Jiangjiang Zhang is pursuing his Doctor degree in Beijing University of Technology, China. His research interest includes data security, privacy protection, computational intelligence, combinatorial optimization and modelling.

Shanshan Tu received his PhD degree from Computer Science Department at Beijing University of Posts and Telecommunications in 2014. From 2013 to 2014, he visited University of Essex for National Joint Doctoral Training. He worked in the Department of Electronic Engineering at Tsinghua University as a postdoctoral researcher from 2014 to 2016. He is currently an Associate Professor and Deputy Dean at Faculty of Information Technology, Beijing University of Technology, China. His research interests are in the areas of cloud computing, MEC and information security techniques.



Hisham Alasmary is an Assistant Professor at King Khalid University. He obtained his Ph.D. from the Department of Computer Science at the University of Central Florida in 2020, and his M.Sc. degree in Computer Science from The George Washington University, in Washington, D.C., USA, in 2016. His research interests include Software Security, IoT Security and Privacy, ML/DL Applications in Information Security, and Adversarial Machine Learning.

MUHAMMAD WAQAS (M'18, SM'22) received his PhD degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2019. From Oct. 2019 to Mar. 2022, he was a Research Associate at the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Currently, he is a Senior Lecturer at the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London, UK. He is also an Adjunct Senior Lecturer at the School of Engineering, Edith Cowan University, Australia. His current research interests are in the areas of Wireless Communication, vehicular networks, cybersecurity and Machine Learning. He is recognised as a Global Talent in the area of Wireless Communications by UK Research and Innovation and a Professional Member of Engineer Australia. He is a senior member of IEEE, a Professional Member of ACM, an IEEE Young Professional, a Member of the Pakistan Engineering Council and PhD approved supervisor by the Higher Education Commission of Pakistan.