Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Lightweight Heterogeneous Signcryption for SDN-IoT Authentication and Access Control

Fahad N. Nife[a], Bilal Majeed Abdulridha Al-Latteef[b]

[a]College of Artificial Intelligence and Cyber Security Engineering, Al-Muthanna University, Al-Muthanna, Al-Samawa, Iraq. Email: fahad.naim@mu.edu.iq

[b]The General Directorate of Qadisiyah Education, Al-Qadisiyah, Al-Diwaniyah, Iraq. Email: bil.pro84@gmail.com

ARTICLE INFO

Article history:
Received: 14/06/2026
Rrevised form: 21/06/2026
Accepted : 22/06/2026
Available online: 30/06/2026

Keywords: Software-Defined Networking, Internet of Things, Heterogeneous Signcryption, Certificateless Cryptography, Identity-Based Cryptography, Authentication, Access Control

ABSTRACT

In modern software defined networking environments, the number of Internet of Things (IoT) devices is rapidly increasing, rendering critical security threats such as authentication and secure communication between heterogeneous cryptographic domains. Most of the existing signcryption schemes are designed in a homogenous cryptographic environment and do not fulfill the needs of interoperability between resource-constraint IoT devices and powerful SDN controllers. This paper presents a new lightweight heterogeneous signcryption scheme facilitating secure bidirectional communication between CLC based IoT devices and IBC based SDN controllers. Our scheme achieves 128-bit security based on elliptic curve cryptography over the secp256r1 curve, and has no computational overhead of bilinear pairings. We provide a complete implementation for Ryu SDN framework, interfaced with Mininet network emulation, illustrating the feasibility of practical deployment. We conduct a comprehensive performance evaluation to demonstrate a throughput of 226.9 ops/s (Signcryption) and up to 2,156.4 ops/s for Optimized Unsigncryption, with linear scalability for more than 100 IoT devices. Formal security analysis through ProVerif confirms that the scheme fulfills confidentiality, authentication, and integrity properties in accordance with the Dolev-Yao attacker model. We also provide mathematical security proofs of IND-CCA2 security and EUF-CMA unforgeability with respect to the Elliptic Curve Discrete Logarithm Problem (ECDLP) and Computational Diffie-Hellman (CDH) assumptions in the Random Oracle Model.

MSC..

https://doi.org/ 10.29304/jqcsm.2026.18.22997

1. Introduction

SDN and the Internet of Things (IoT) have changed not only how we approach network security, but also the way in which we store data. The problem of protecting communication between known ends, which used to be a much simpler one, has become something infinitely more complicated [1]. The modern networks are required to accommodate billions of heterogeneous devices which largely differ in terms of computational power and security requirements. The world is expected to have more than 29 billion connected IoT devices by 2030 and many of them

∗Corresponding author: Fahad Naim Nife

Email addresses: fahad.naim@mu.edu.iq

Communicated by ‘sub etitor’

will be used in critical infrastructure, health care, and industrial environments [2]. A hack on a smart meter, or an unsolicited command to an industrial actuator could propagate to fully disastrous crashes. Nevertheless, even with this time-sensitivity, most SDN-IoT environments implement security strategies that either compromise the urgency of every component's action, or, employ resource-intensive protocols that cannot be executed by devices bound by higher operational constraints.

This ideal state would employ a transparent security model where IoT devices authenticate in real time with SDN controllers, send data covertly and sign messages while keeping their analogous processing power from being stopped by heavy weight cryptography. Furthermore, in this vision a smart factory temperature sensor would send its readings to a controller which by itself issues encrypted commands for actuators, with each part (sensor, controller, actuator) cryptographically assuring the identity of the others involved. The truth is basic stress between the power of security and also the practicality of miscalculation.

Public Key Infrastructure PKI is based on known standards; however, it imposes certificate management costs that are not sustainable at IoT scale. Identity-Based Cryptography (IBC) [3] is a nice alternative where the public key of each user is an identity string, which has been realized in practice by Boneh and Franklin [4]. IBCs unfortunately suffer from the key escrow problem; a Key Generation Center (KGC) is able to compute all private keys, and thereby impersonate any user. In order to solve these problems, Al-Riyami and Paterson [5] introduced Certificateless Cryptography (CLC), which gives the contribution of the user to private key generation so that no party can get the complete secret. But CLC and IBC are fundamentally incompatible cryptographic domains. A CLC-based IoT device cannot communicate directly with an IBC-based controller using standard protocols where the mathematical structures on which their underlying encryption processes rest are categorically non-interoperable.

The heterogeneity problem is not just a theoretical one. Due to heterogeneous nature of third-party network devices and controllers in many real SDN-IoT deployments, they implement different cryptographic framework. As an example, a manufacturer of industrial sensors might choose CLC as they are resistant to key escrow tremendous risk, whereas an SDN controller vendor would use IBC thanks to the easy handling of keys. There are natural limits to diversity and forcing homogeneity is both impractical and undesirable.

Many research efforts have tried to overcome this challenge. Li et al. [6] introduced a signcryption scheme for wireless sensor networks with fair efficiency, however directly works only in PKI domain. Zhang et al. [7] proposed an identity-based key agreement protocol, which is based on bilinear pairings; which, due to operating with bilinear pairings (a computationally expensive operation), makes the scheme impractical for resource-limited devices. Sun et al. [8] proposed a certificateless signcryption scheme that is more efficient, but the scheme demands that all parties reside in the same cryptographic domain. Karati et al. [9] introduced a certificateless signature scheme for Industrial IoT, which also relies on pairing computations and does not simultaneously ensure confidentiality and authenticity. Yet the gap remains: It is known that no lightweight and pairing-free signcryption scheme can securely be used for bidirectional communication between CLC-based IoT devices and IBC-based SDN (Software Defined Network) controllers. This gap has serious consequences. This forces organizations into a hard choice, adopting homogenous solutions that do not meet security needs or deploying cumbersome protocol translation layers, which add additional latency and attack surfaces. This restriction hampers SDN in environments where security is paramount, exactly where software-defined management would be most effective.

The theory for tackling this problem is exist. Zheng [10] proposed a new primitive called signcryption, which showed that confidentiality and authenticity can be achieved at the same time at less cost than the two separate operations of encryption and signature. Elliptic Curve Cryptography (ECC) is based on the works of Koblitz [11] and Miller [12], providing equivalent security to RSA with orders of magnitude smaller key sizes. Signcryption scheme can be built upon elliptic curves without needing bilinear pairings, attaining strong security guarantees and remaining computationally reasonable for constrained devices.

In this paper, we fill the gap by presenting a novel lightweight heterogeneous signcryption scheme for SDN-IoT authentication and access control policy. We propose a lightweight scheme providing bidirectional secure communication between CLC-based IoT devices and IBC-based SDN controllers, free of computationally expensive bilinear pairings. This method uses the NIST P-256 elliptic curve to provide 128-bit security and executes efficiently on resource-constraint appliances. We implement the cryptographic protocol within Ryu SDN framework and evaluate practical feasibility in Mininet simulation network. Formal security analysis employing ProVerif [13] proves that the scheme satisfies confidentiality, authentication and integrity under Dolev-Yao attacker model [14].

The contributions of this work are fourfold: (1) establishment of a heterogeneous signcryption scheme that connects CLC and IBC domains with bilinear pairing overhead removed; (2) formal security proof establishing IND-CCA2 confidentiality and EUF-CMA unforgeability; (3) development of a complete prototype for seamless integration with SDN infrastructure; and finally (4) thorough performance evaluation demonstrating linear scalability in terms of device count.

This work has implications for both the academic and applied domains. At the theoretical level we contribute positively to the heterogeneous cryptography literature by showing that secure cross-domain signcryption need not involve computational trade offs. The scheme suits instant needs for manufacturing securing SDN-IoT areas, but also establishes a practical common standard how to achieve device diversity security.

The remaining parts of this paper are organized as follows. Section 2 surveys related work. Section 3 presents mathematical preliminaries. The system model and their security requirements are shown in Section 4. Section 5 describes the proposed scheme, followed by a security analysis in section 6. Implementation and performance evaluation are discussed in Section 7. Section 8 addresses the limitations and Section 9 concludes the paper.

2. Related Work

This part is to review state-of-the-art literature among four interrelated and common areas including SDN security solutions, IoT authentication protocols, signcryption schemes and heterogeneous cryptosystems. By synthesizing these bodies of work, we identify the concrete knowledge gap that underlies our research and locate how our contribution relates to existing scholarly literature.

2.1. SDN Security Mechanisms

Software-Defined Networking (SDN) has a centralized architecture, which creates both opportunities and vulnerabilities for security management. Kreutz et al. [15] conducted a study of major SDN threats and vulnerabilities, identifying seven threat vectors including unauthorized controller access, impostor flow rules and DoS attacks targeting the control plane. Their taxonomy is comprehensive and still serves as a common point on the topic; however, this work occurred before IoT devices were widely integrated into systems, and hence does not address authentication challenges associated with resource-constrained endpoints.

Ahmad et al. [16] reviewed SDN security by analyzing 89 studies to group its threats and countermeasures. Their study found that most of the proposed solutions assume a homogeneous device capability, which is a huge limitation given that IoT devices with kilobytes memory need to create interactions with powerful controllers. Similarly, Scott-Hayward et al. [17] recognized the controller as single point of failure and suggested distributed architecture, whereas their security mechanisms were based on standard TLS handshakes that are not appropriate for constrained devices. Cumulatively these studies indicate that SDN security research has been insufficiently anchored in the heterogeneous computational landscape of IoT environments.

2.2. IoT Authentication Protocols

In IoT contexts, authentication must use protocols that achieve the strength of security relating to computational limitations. Ferrag et al. Presenting a comprehensive survey of authentication protocols for IoT, [18] classified those according to their cryptographic method of operations into three schemes: symmetric key; public key; and hybrid. They discovered an intrinsic tradeoff: symmetric schemes are efficient but also vulnerable in the key distribution context whereas public be strategies are much more dependable but at insurmountable computational expense. To note, none of the analyzed protocols tried cross-domain authentication for different cryptographic systems.

Wazid et al. In [19], they proposed a hash and XOR only based lightweight authentication scheme for the smart grid environment. Their scheme provides no confidentiality, relies on the use of pre-shared secrets (which complicates device provisioning) and while it has decent performance (authentication times in the millisecond range). Challa et al. In [20], authors proposed to ECC-based authentication protocol for healthcare IoT and showed elliptic curve operations can be performed on constrained devices. Their protocol consists of three exchanges that result in latency unsuitable for real-time industrial applications. While both the studies show the potential of lightweight cryptography for Internet of Things, neither considers either signcryption paradigm or heterogeneous environments.

2.3. Signcryption Schemes

Signcryption, which was put forward by Zheng [10], is a new encryption model that can change our common perception of the confidentiality and authenticity in the meantime. Signcryption combines signing and encryption into a single logical step instead of a sequence of sign-then-encrypt, achieving lower computational and communication costs. That increase of operational efficiency is particularly useful within IoT contexts.

Baek et al. [21] provides a concrete definition for sampling security of signcryption, where standard notions are IND-CCA2 and EUF-CMA. Their definition-based framework, however, is still only applicable to homogeneous cryptographic domains. In [22], again Malone-Lee laguages the privided design as a signcryption, by surrounding the identity-based form. Its construction uses bilinear pairings, an operation that requires 20x more work than essentially performing vice-free elliptic curve scalar multiplication [23]. However, that overhead is unsustainable for low-resources IoT devices.

Zhou [8] presented a certificateless signcryption without random oracles (standard model), which consolidated the theoretical framework of certificateless signcryption, at the same time still stayed in one cryptographic domain (CLC). Li et al. [24] proposed a certificateless hybrid signcryption technique for better performance, but it is still single-domain construction and does not support heterogeneous cross-domain communication.

2.4. Heterogeneous Cryptographic Systems

Heterogeneous signcryption constructions arise from the need to enable secure communication across different public-key paradigms (e.g., PKI, certificateless and identity-based). Early heterogeneous signcryption foundations emphasized additional privacy goals, e.g., key privacy, while providing cross-domain confidentiality and authenticity, but those designs commonly utilize bilinear pairings to achieve identity-based functionality which becomes a bottleneck on resource constraint IoT endpoints [25].

To address online latency or offload computation from resource-limited devices, heterogeneous online/offline signcryption was proposed for IoT communication. Nevertheless, leading heterogeneous online/offline and CLPKC→IDPKC-style heterogeneous signcryptions remain pairing-based [38], [39] so they inherit the bilinear mappings cost of phases through online phase, or/and through entire signcryption/designcryption processing.

More recent IoT-oriented work has also tended toward pair-free heterogeneous designs and pragmatic deployment assumptions (e.g., edge-assisted environments). An example of this is a generic heterogeneous pairing-free signcryption construction for IoMT/edge settings [13] showing an ongoing interest in heterogeneous schemes that suit better limited devices and gateway/controller architectures [40]. Moreover, heterogeneous generalized signcryption has been explored in NDN-enabled IoT scenarios featuring lightweight curve primitives with anti-replay and forward secrecy articulated as clear design objectives along with automated verification (AVISPA [41]).

Besides classical security properties like confidentiality and authenticity, some heterogeneous schemes also provide auxiliary or functionality such as equality testing which may better suit the needs of an IoT/SDN data analytics workflow but at extra design trade-offs and added attack surfaces; heterogeneous signcryption with equality test (e.g., IBC→PKI) is representative work in this direction [26].

2.5. Critical Synthesis and Gap Analysis

Analysis of this literature demonstrates a number of trends and evidences a knowledge gap. First, the SDN security research has become mature but still assumes homogeneous devices. Second, while IoT authentication protocols are promising in terms of efficiency, they seldom offer built-in confidentiality. Third, signcryption schemes yield the necessary combination of guarantees but either incur prohibitive pairing operations or assume a homogeneous domain. Fourth, heterogeneous cryptographic constructions have been proposed but they are not verified in practical SDN-IoT operations. However, there is still a gap between (i) heterogeneous signcryption that fits SDN–IoT realities (controller/gateway vs. constrained endpoints), and (ii) end-to-end validation within SDN environments (e.g., controller integration, realistic traffic/emulation, and clear replay/freshness handling). Table 1 summarizes the heterogeneous baselines most similar to our target setting and compares what each work explicitly reports in terms of proof model, replay/freshness mechanism, and implementation evidence. We chose the comparison baselines based on three criteria: (1) heterogeneous signcryption (cross domain, not heterogeneous), (2) involving at least one of CLC or IBC, and (3) published in peer-reviewed venues by 2024. The homogeneous schemes and

general-purpose signcryption constructions were omitted because they are not suitable for the purpose of this work, namely the cross domain interoperability challenge. We recognize that the signcryption landscape is still evolving and that new constructions that feature post-quantum primitives or attribute-based policies could be compared in the future when they are more developed.

Table 1 - Comparison with Existing Schemes.

Scheme	Heterogeneous direction	Pairing usage	Proof model / assumptions (as stated)	Formal verification	Replay / freshness mechanism (as stated)	Real deployment / prototype
Proposed scheme (this work)	CLC → IBC	Pairing-free (ECC scalar mult + hashes)	IND-CCA2 + EUF-CMA proofs in ROM; sender forward secrecy claimed (see note in Table V)	ProVerif validation is included	Timestamp + cache/checklist anti-replay described	SDN-IoT prototype described (Ryu controller + Mininet)
HOOSC (heterogeneous online/offline signcryption for IoT)	CLC → PKI	Pairing-based (bilinear map (e))	IND-CCA2 + EUF-CMA security theorems stated (ROM; BDHI / q-SDH mentioned)	Not reported in the paper excerpted	Not explicitly described in the excerpted sections	Not reported in the excerpted sections
Niu et al. "Efficient heterogeneous signcryption from CLPKC to IDPKC"	CLPKC → IDPKC	Pairing-based (pairing computations counted in performance)	Claims IND-CCA + EUF-CMA in ROM (stated in abstract)	Not reported in the excerpted sections	Not reported in the excerpted sections	Not reported in the excerpted sections
PK-CLET (heterogeneous signcryption w/ equality test)	PKI ↔ CLC (heterogeneous)	Pairing-based (pairing cost counted)	Claims IND-CCA2 and EUF-CMA in ROM (summary statement)	Not reported in the excerpted sections	Not reported in the excerpted sections	Not reported in the excerpted sections
Kasyoka et al. (IoMT + edge, pairing-free heterogeneous signcryption)	Heterogeneous (IoMT setting)	Pairing-free (explicit motivation vs pairings)	States IND-CCA2 + EUF-CMA (Type-I/II adversary)	Not reported in the excerpted sections	Not reported in the excerpted sections	Not reported in the excerpted sections
HGSC (NDN-IoT, hyperelliptic curve)	CLC → IBC	Pairing-free motivation; highlights bilinear pairing overhead	Threat model + properties enumerated; forward secrecy + anti-replay defined as goals	AVISPA validation claimed	Anti-replay via nonce/timestamp techniques is described	Practical NDN smart-city scenario claimed

Table 1 compares our work with the nearest heterogeneous signcryption baselines. In particularly, it outperforms against pairing-based construction (HOOSC, Niu et al. [39], and PK-CLET), our model circumvents the bilinear pairing bottleneck, with equal or stronger formal security guarantees.

A key differentiator arises in the replay and freshness dimension: out of the six schemes considered, only our scheme and HGSC employ explicit anti-replay mechanisms. While HOOSC, Niu et al. [39], and Kasyoka et al. employ ephemeral randomness to provide semantic security, do not bind ciphertexts to any specific time window and thus can potentially be replayed in the high-frequency SDN environments.

In contrast, our scheme adopts a multi-layered defence: it integrates timestamp-based freshness with a controller-side replay cache indexed by (IDD, R) tuples. This will provide the assurance that even if a replayed ciphertext arrive within valid timestamp window (Δ), it will be identified. In addition, our work is also the first one targeting to validate both those properties using ProVerif and then produce an operational prototype specifically designed for SDN requirements with verified emulation results on a network, thus bridging theoretical protocol design and practical SDN orchestration.

3. Preliminaries

This part presents the notation and cryptographic background needed for understanding the proposed scheme. We take elliptic curve arithmetic knowledge for granted and talk about only the concrete constructions that matter to our heterogeneous signcryption protocol.

3.1. Notation

Let $\kappa = 128$ be the security parameter. Let $\mathbb{G}$ be an elliptic-curve group of prime order $q$generated by P. Scalar operations are over $\mathbb{Z}_q^*$. For any scalar $a \in \mathbb{Z}_q^*$, aPdenotes elliptic-curve scalar multiplication. The concatenation operator is denoted by $\|$.

Hash/KDF functions (domain-separated). To prevent hash outputs from being reused for various reasons (binding identity, signature challenges and key derivation), we model three independent primitives and instantiate them using domain separation:

- $H_s(\cdot): \{0,1\}^* \to \mathbb{Z}_q^*$(hash-to-scalar), used for identity scalars and signature challenges.
- $H_k(\cdot): \{0,1\}^* \to \{0,1\}^{256}$(key derivation), used to derive symmetric session keys.
- $H_n(\cdot): \{0,1\}^* \to \{0,1\}^{96}$(nonce derivation), used only if a deterministic AEAD nonce is needed (otherwise a random 96-bit nonce is sampled).

Hash/KDF functions (domain-separated). To prevent hash outputs from being reused for various reasons (binding identity, signature challenges and key derivation), we model three independent primitives and instantiate them using domain separation:

- $H_s(\cdot): \{0,1\}^* \to \mathbb{Z}_q^*$(hash-to-scalar), used for identity scalars and signature challenges.
- $H_k(\cdot): \{0,1\}^* \to \{0,1\}^{256}$(key derivation), used to derive symmetric session keys.
- $H_n(\cdot): \{0,1\}^* \to \{0,1\}^{96}$(nonce derivation), used only if a deterministic AEAD nonce is needed (otherwise a random 96-bit nonce is sampled).

Instantiation (implementation guidance). H_s is implemented as SHA-256 followed by integer reduction modulo q, with rejection sampling if the result is 0. H_k is instantiated using HKDF-SHA256 to output 32 bytes. Domain separation is realized by prefixing a short ASCII tag (e.g., "HS", "HK", "HN") to the input before hashing/KDF.

Important note (pairing-free / no hash-to-curve). This work does **not** require any hash-to-curve (MapToPoint) operation. All identity processing uses hash-to-scalar H_s, which is computationally lightweight and consistent with the pairing-free design goal.

Table 2 summarizes the notation used throughout this paper.

Table 2 - Notation.

Symbol	Description
κ	Security parameter (κ = 128)
$\mathbb{G}$	Elliptic curve group of prime order q
P	Generator of $\mathbb{G}$
q	Order of $\mathbb{G}$ ($\approx 2^{256}$)
$\mathbb{Z}q^*$	Multiplicative group of integers modulo q
s, s_m	Master secret keys (IBC, CLC respectively)
P_{pub}, P_m	Master public keys (IBC, CLC respectively)
ID	Identity string
d	IBC private key
Q	IBC public key derived from identity
(D, x)	CLC private key (partial key, secret value)
(R, X)	CLC public key components
$H_s : \{0,1\}^* \to Zq^*$	hash-to-scalar (SHA-256 mod q, domain-separated)
$H_k : \{0,1\}^* \to \{0,1\}^{256}$	KDF (HKDF-SHA256)
H_n:$\{0,1\}^* \to \{0,1\}^{96}$	nonce derivation (domain-separated)
m	Plaintext message
σ	Signcrypted ciphertext
‖	Concatenation operator
←$	Random sampling

3.2. System Model

1) Elliptic Curve Cryptographic Setting

We work over NIST P-256 elliptic curve (secp256r1), which gives estimated 128-bit security level [28]. All elliptic curve group operations, such as scalar multiplication and point addition are performed per secp256r1 parameters.

2) Identity-Based Cryptosystem (IBC Domain, Pairing-Free)

At the controller domain we employ an identity-based key generation mechanism where public keys are generated from identity strings without using certificates. A trusted IBC-KGC selects a master secret $s \leftarrow \$ \mathbb{Z}_q^*$ and publishes $P_{\text{pub}} = sP$.

For an entity with identity ID, define the identity scalar

$$h_{ID} = H_s(\text{"ID"} \parallel ID) \in \mathbb{Z}_q^*.$$

The entity's **implicit public key** is point on the curve

$$Q_{ID} = h_{ID} \cdot P_{\text{pub}} = (s \cdot h_{ID})P.$$

The IBC-KGC assigns and then issues the matching private key acting as a scalar

$$d_{ID} = s \cdot h_{ID} \, mod \, q.$$

Because this pairing-free IBC formulation enables certificate-free public keys derived from identity and standard elliptic-curve scalar multiplication. It is a heuristic choice which helps in constructing lightweight heterogeneous signcryption without bilinear pairings.

3) Certificateless Cryptosystem

To overcome the key escrow problem of IBC, Certificateless cryptography (CLC) distributes the key generating task between KGC and user [5]. The KGC generates a partial private key D using its master secret s_m, while the user selects a secret value $x \in \mathbb{Z}_q^*$, independently. The resulting private key is $SK = (D, x)$, and the corresponding public key is $PK = (R, X)$, where R is generated during partial key extraction and $X = xP$. This construct ensures that neither the KGC nor an external adversary may totally compromise the user's private key.

4) Signcryption

Signcryption is a cryptographic primitive that combines digital signature and encryption into one logical operation, achieving confidentiality and authenticity at the same time [10]. Signcryption achieves equivalent security guarantees while significantly reducing computational overhead and communication cost relative to traditional sign-then-encrypt approaches.

3.3. Hardness Assumptions

Security of our scheme boils down to two well-studied problems.

Definition 1 (ECDLP). Given $P, Q \in \mathbb{G}$ where $Q = xP$ for unknown $x \in \mathbb{Z}_q^*$, computing x is infeasible for any probabilistic polynomial-time adversary.

Definition 2 (CDH). Given $P, aP, bP \in \mathbb{G}$ for unknown $a, b \in \mathbb{Z}_q^*$, computing abP is infeasible for any probabilistic polynomial-time adversary.

These assumptions are standard in elliptic curve cryptography, and have withstood decades of cryptanalytic attack on curves of sufficient size [23].

3.4. Security Definitions

We adopt the standard security notions for signcryption formalized by Baek et al. [21].

Definition 3 (IND-CCA2). A signcryption scheme achieves indistinguishability under adaptive chosen-ciphertext attack if no adversary, given access to an unsigncryption oracle, can distinguish between signcryptions of two chosen messages with probability non-negligibly better than 1/2. The adversary may query the oracle on any ciphertext except the challenge.

Definition 4 (EUF-CMA). A signcryption scheme achieves existential unforgeability under chosen-message attack if no adversary, given access to a signcryption oracle, can produce a valid signcryption for any new message with non-negligible probability.

Definition 5 (Sender Forward Secrecy). A signcryption scheme provides sender forward secrecy if compromise of the sender's long-term private key does not enable an adversary to decrypt previously captured ciphertexts generated by that sender. The session keys must depend on ephemeral secrets that are erased after each signcryption.

3.5. Design Rationale

Our construction differs from earlier work in several design decisions. First, we never use bilinear pairings at all. Despite enabling elegant constructions, one pairing operation costs around 20× more than scalar multiplication [23] preventing deployment on constrained IoT devices. Second, we enable heterogeneous communication—CLC devices to communicate with IBC controllers—without protocol translation layers. Through careful key derivation interacting both sides the scheme unifies the mathematical structures. Third, we add sender forward secrecy by means of ephemeral Diffie-Hellman values generated by the sender for each signcryption. Thus, compromising of the long-term private key of an IoT device will not expose any previously transmitted ciphertexts generated by that IoT device.

4. System Model and Security Requirements

This section describes the network architecture, the participating entities, threat model and security objectives which serve as a guideline for our scheme design.

4.1. Network Architecture

The SDN-IoT architecture consists of two cryptographic domains linked by the proposed heterogeneous signcryption protocol, as shown in Fig. 1.

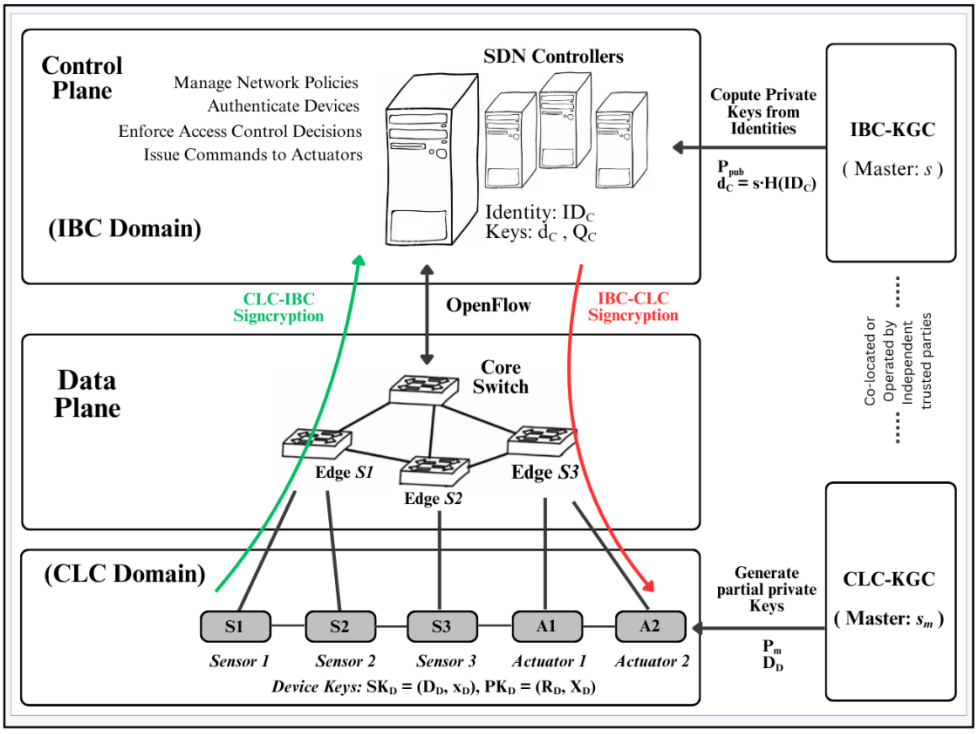

Fig. 1 - SDN-IoT Heterogeneous Signcryption Architecture.

The control plane is composed of one or several SDN controllers present in the IBC domain. These controllers are able to manage network policies, authenticate devices, enforce access control decisions, and send commands to actuators. The IBC framework makes controller key management easier, controller identities (such as: Controller_Primary) become public keys and alleviates the burdensome complexity of provisioning certificates.

The data plane includes IoT devices working in the CLC domain. Such as resource limited sensors sending readings of physical environments and actuators receiving commands for control. CLC offers key escrow resistance needed for deployments involving a distinct separation of device manufacturers and network operators, each with different trust assumptions. Devices communicate through switches that are OpenFlow-enabled and forward authentication requests and encrypted data between planes.

The IBC-KGC and the CLC-KGC are two Key Generation Centers that manage their respective domain; the former maps identities to controller private keys, while the latter generates partial private keys for devices. These KGCs may be co-located or operated by independent trusted third parties depending on deployment requirements.

4.2. Entity Definitions

SDN Controller ($\mathcal{C}$). The controller has identity ID_C, the private key d_C derived from the IBC-KGC and implicitly defined public key $Q_C = H(ID_C)\,P$; it authenticates every incoming device request while managing an access control list and processes measurements received from sensors before signcrypting commands to authorized actuators.
IoT Device ($\mathcal{D}$). Each device has its identity ID_D, its full private key $SK_D = (D_D, X_D)$which is composed from the CLC-KGC's partial key and a self-generated secret, as well as public key $PK_D = (R_D, X_D)$. Sensors signcrypt telemetry data to the controller and actuators unsigncrypt received commands, execution of actions authorized.
IBC-KGC. Trusted authority with master secret s. It extracts private keys for the controllers upon identity registration. With Assuming honest and secure throughout the system lifetime.
CLC-KGC. Trusted authority with master secret s_m. It generates part of the private keys for the devices. But it cannot impersonate them without knowledge of user-generated secret values.

4.3. Threat Model

We use Dolev–Yao adversary model [14], which is a standard for cryptographic protocol analysis. The adversary $\mathcal{A}$ has access to the following abilities:

1. **Network Control.** $\mathcal{A}$ totally controls the communication channel—monitoring all transmissions, intercepting messages, injecting forged ones, replaying captured messages and reordering or deleting legitimate traffic.
2. **Device Compromise.** $\mathcal{A}$ may sequester a subset of IoT devices, and acquire complete private keys for these devices. A compromised device can be involved in active attacks against any uncompromised entity.
3. **Computational Bounds.** $\mathcal{A}$ is probabilistic polynomial-time bounded. It is intractable to solve the ECDLP, or the CDH on this curve.

Trusted Components. An assumption is that both KGCs and the master secrets stored in them, as well as SDN controller are not compromised. This simulates realistic deployments, where controllers will be in tape and physically secured datacenters with practices already established. Relaxing these assumptions (e.g., distributed controllers, threshold KGCs) remains future work. The trust assumption can be reduced in decentralized deployments, where one trusted KGC is not possible, by using (t, n) threshold secret sharing whereby the master secret is shared among n parties, and t of them are required to reveal the private keys. It helps to prevent any one compromised KGC node from impersonating a controller or a device. In multi-vendor ecosystems, the device manufacturers can have their own CLC-KGCs during the manufacturing process and network operators can have their own IBC-KGCs for controllers, thereby establishing trust boundaries along the existing organizational boundaries. The split key generation of the CLC domain naturally reduces the risk of a single compromised CLC-KGC as it will not be able to impersonate a device without the user-generated secret value x_D.
Out of Scope. Side-channel attacks, implementation vulnerabilities, and denial-of-service attacks are outside our threat model. We focus on cryptographic protocol security.

4.4. Security Requirements

The proposed scheme has to satisfy the following properties:
SR1: Confidentiality. Only the intended recipient can decrypt signcrypted messages. For an adversary that sees the ciphertexts, they learn nothing about plaintext, only the length of messages.
SR2: Authentication. The receiver verifies the identity of sender through cryptographic guarantees. Impersonation is impossible without knowledge of their private key.
SR3: Integrity. In unsigncryption, any change in ciphertext or associated data is detected. Tampered messages are rejected.
SR4: Non-repudiation. Signcrypted messages cannot be denied by the sender. The signature is embedded, tying sender identity to message contents.
SR5: Sender Forward Secrecy. The compromise of an IoT device's long-term private key does not allow the decryption of previously captured ciphertexts sent by that device. Every signcryption carries with g a new ephemeral secret erased after the use.
SR6: Replay Resistance. The scheme is a secure against replay attack signcryption. Each ciphertext includes a new ephemeral value R and the current timestamp T. The receiver keeps a replay cache of accepted (R, ID_D) pairs that are fresh within some window Δ, and rejects any ciphertext including a pair that has already been seen or an expired timestamp.

4.5. Design Goals

Beyond the basic requirement of security, practical deployment introduces additional necessities:
DG1: Lightweight Computation. Operations have to execute well on constrained devices. We aim for sub-10ms signcryption on typical IoT hardware, achieved by avoiding bilinear pairings.
DG2: Minimal Communication. Signcryption should require single-round transmission. Multi-round handshakes cause latency lag and consume more energy for battery-powered devices.
DG3: Heterogeneous Interoperability. CLC devices and IBC controllers communicate without the need for protocol translation, gateway proxies or conversion of a cryptographic domain.
DG4: Scalability. As more devices are added, performance should degrade linearly rather than exponentially. Hundreds of devices need to authenticate at the same time.
DG5: Standard Compliance. The scheme uses NIST-approved primitives (namely P-256, SHA-256 and AES-256-GCM) ensuring the regulatory acceptance while also utilizing the hardware acceleration that is widely available on modern IoT chipsets.
Together, these requirements and goals characterize the design space for our heterogeneous signcryption scheme. Section V presents the concrete construction that meets all requirements.

5. Proposed Scheme

In this section, we describe our lightweight heterogeneous signcryption scheme to facilitate secure bidirectional communication between CLC-based IoT devices and IBC-based SDN controllers. We begin with a description of the system initialization, and next move on to describe core signcryption algorithms.

5.1. System Initialization

IBC Domain Setup. The IBC-KGC selects a master secret $s \leftarrow \$\, \mathbb{Z}_q^*$ and publishes $P_{\text{pub}} = sP$. For a controller with identity ID_C, compute $h_C = H_s(\text{"ID"} \parallel ID_C) \in \mathbb{Z}_q^*$. The controller's private key is $d_C = s \cdot h_C \, mod \, q$, and its implicit public key is $Q_C = h_C \cdot P_{\text{pub}} = d_C P$. Any party can compute Q_C from (ID_C, P_{pub}) without certificates.

CLC Domain Setup. The CLC-KGC selects master secret $s_m \leftarrow \$\, \mathbb{Z}_q^*$ and publishes $P_m = s_m \,.P$. Device registration proceeds as follows: for device identity ID_D, the KGC selects $r_D \leftarrow \$\, \mathbb{Z}_q^*$, computes $R_D = r_D \,.P$ and partial private key $D_D = r_D + s_m \,.H_s(\text{"CLC"} \parallel \text{ID}_D \parallel \text{R}_D) mod \, q$.

The device independently generates secret value $x_D \leftarrow \$\, \mathbb{Z}_q^*$ and computes $x_D = x_D \,.P$. The complete key pair is $SK_D = (D_D \,, x_D)$ and $PK_D = (R_D \,, x_D)$. Public key validity is verifiable: $D_D \,.\, P = R_D + H(\text{ID}_D \parallel \text{R}_D) \,.P_m$.

Key Distribution. During deployment, controllers receive their corresponding private keys from the IBC-KGC over secure channels. IoT devices must register with the CLC-KGC when they are manufactured or provisioned. Cross-domain public parameters $(P_{pub} \,, P_m)$ are shared to all entities. Device public keys PK_D are sent to controllers on initial authentication and cached for future communications.

5.2. CLC-to-IBC Signcryption

This allows for an IoT device to transfer verified, encrypted data to the SDN controller. This construction combines confidentiality via ECDH key agreement with the IBC master public key, and authenticity via Schnorr-style signatures all in a single operation.

Algorithm 1: CLC-to-IBC Signcryption
Input: Message m, sender private key $SK_D = (D_D, x_D)$, sender public key $PK_D = (R_D, X_D)$, receiver identity ID_C, IBC master public key P_{pub}, timestamp T
Output: Signcryption σ
1: r ←$ $\mathbb{Z}q^*$
2: R ← r·P
3: h_C ← Hs("ID" ‖ ID_C)
4: Q_C ← hC · P_{pub}
5: S ← r·Q_C // Shared secret: r·H(ID_C)·s·P
6: K ← H(S ‖ ID_C ‖ T) // Session key derivation
7: N ←$ $\{0,1\}^{96}$
8: (C, tag) ← AES-GCM.Enc(K, N, m, ID_C)
9: h ← Hs("CH" ‖ C ‖ R ‖ T ‖ ID_C ‖ PK_D)
10: σ_{sig} ← r + (D_D + x_D)·h mod q
11: return σ = (R, C, N, tag, σ_{sig}, T, PK_D)

Correctness. The receiver holding $d_C = s.H(ID_C)$ computes the shared secret as $S' = d_C . R = s.H(ID_C).r.P$. The sender computes $S = r . Q_C = r . H(ID_C).P_{pub} = r . H(ID_C).s . P$. Since scalar multiplication is commutative, $S = S'$, enabling consistent session key derivation and successful decryption.

Design Rationale. Line 3 calculates the controller's implicit public key $Q_C = H(ID_C).P_{pub}$, where the IBC master public key is $P_{pub} = s . P$, guaranteeing that the shared secret contains a component of the master secret factor s thereby enabling the receiver to compute a matching version of the shared secret just with its private key d_C. Lines 8-9 bind the signature to ciphertext, timestamp and sender's public key to avoid mauling attacks. Line 5 timestamp T ensures session key freshness. We stress that the property that is obtained is sender forward secrecy (Definition 5), which protects against sender key compromise, but is not perfect forward secrecy under receiver key compromise (see Section 8.2).

5.3. CLC-to-IBC Unsigncryption

This algorithm is performed by SDN controller when it receives signcrypted data from an IoT device.

Algorithm 2: CLC-to-IBC Unsigncryption
Input: Signcryption σ = (R, C, N, tag, σ_{sig}, T, PK_D), receiver private key d_C, receiver identity ID_C, sender identity ID_D, maximum age Δ
Output: Message m or ⊥
1: if $|T_{current} - T| > \Delta$ then return ⊥
2: if ReplayCache.contains(ID_D, R) then return ⊥
3: Parse $PK\_D = (R_D, X\_D)$
4: h_D ← Hs("CLC" ‖ ID_D ‖ R_D)
5: PK_{full} ← R_D + h_D·P_m + X_D
6: h ← H_s("CH" ‖ C ‖ R ‖ T ‖ ID_C ‖ PK_D)
7: if σ_{sig}·P ≠ R + h·PK_{full} then return ⊥
8: S ← d_C·R
9: K ← H_k("KDF" ‖ S ‖ ID_C ‖ T)
10: m ← AES-GCM.Dec(K, N, C, tag, ID_C)
11: if decryption fails then return ⊥
12: ReplayCache.insert(ID_D, R, T)
13: return m

Verification Logic.
Line 6 verifies the signature by checking $\sigma_{sig} . P = R + h . PK_{full}$. Expanding the signature: $\sigma_{sig} . P = (r + (D_D + x_D) . h) . P = rP + h . (D_D + x_D) . P = R + h . (D_D . P . X_D)$. Since $D_D . P = R_D + h_D . P_m$ (from CLC key validity), we obtain $R + h . (R_D + h_D . P_m + X_D) = R + h . PK_{full}$, confirming correctness.

Security Properties Achieved.
This combination of timestamp validation and a controller-side replay cache prevents replay attacks. The freshness of each message is checked with a timestamp, while the replay cache prevents acceptance of each ephemeral value R

more than once within the freshness window Δ which rules out delayed and fast replays. Authentication and integrity are validated with signature verification (line 6). AEAD decryption (line 9) ensures confidentiality along with ciphertext integrity. The ephemeral r is never reused, and that the session key K depends on the fresh timestamp, allowing for forward secrecy.

5.4. IBC-to-CLC Signcryption

Basically, the communication from SDN controller to IoT devices is quite similar but changes according to their roles. Commands are signcrypted by the controller with its IBC private key d_C and encryption to the device's CLC public key.
The key differences are:

1. **Shared Secret Computation.** The controller computes $S = r \,.\, PK_{full}$ where $PK_{full} = R_D + H(ID_D \parallel R_D) \,.\, P_m + X_D$, utilizing both CLC public key components.
2. **Signature Generation.** The signature becomes $\sigma_{sig} = r + d_C \,.\, h \bmod q$, using the IBC private key structure.
3. **Ciphertext Contents.** The output includes the controller's identity ID_C rather than a public key, as IBC public keys are identity-derived.

The device unsigncrypts by computing $S = (D_D + x_D) \,.\, R$ using its full CLC private key and verifying $\sigma_{sig} \,.\, P = R + h \,.\, Q_C$ where $Q_C = H(ID_C) \,.\, P$.

5.5. Protocol Integration

Fig. 2 show the authentication and data flow of both signcryption directions.

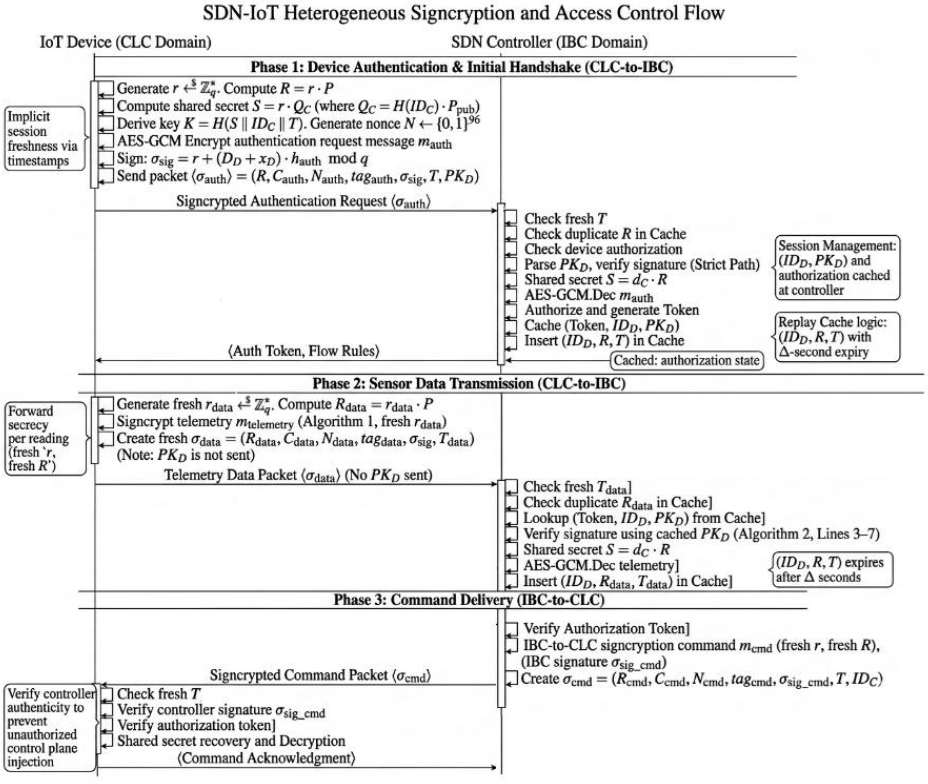

Fig. 2 - Protocol Flow: Authentication and Secure Data Transmission.

Phase 1: Device Authentication. Using Algorithm 1, the IoT device signcrypts an authentication request that includes its identity and capabilities. The controller unsigncrypts through Algorithm 2, checks device authorization based on access control policies, and stores the public key PK_D for future communications.

Phase 2: Sensor Data Transmission. Authenticated sensors signcrypt telemetry readings to the controller with specified periodicity. Here, each transmission utilizes a new ephemeral key r and timestamp ensuring forward secrecy across reads.

Phase 3: Command Delivery. The controller signcrypts commands to authorized actuators via IBC-to-CLC signcryption. To prevent injection into the control plane from being executed, actuators authenticate the authenticity of a controller before issuing commands.

Session Management. After authentication, device public keys are cached at the controller so they do not need to be re-transmitted repeatedly. Since timestamps provide an implicit session freshness without the need to maintain any explicit state, this reduces controller memory overhead needed for tracking thousands of devices.

Replay Cache Management. To avoid replay attacks in the timestamp freshness window, the SDN controller keeps a simple replay cache that stores (ID_D , R) pairs for signcryptions just accepted.

Each entry, associated with its timestamp T will expire automatically after Δ seconds. Each R is randomly generated elliptic-curve point for a unique signcryption, so the cache size can easily be made linear with respect to the number of active devices and message rate. In practice the cache can be based on a hash table or Bloom filter with insignificant memory overhead. Each entry contains about 97 bytes, 32 of which are used for the device identity hash, 33 bytes for the compressed ephemeral point R, and 32 bytes for the timestamp and metadata. The maximum cache size is N × f × Δ entries for a deployment of N devices transmitting at rate f messages per second having freshness window Δ. The representative case of 10,000 devices at 1 msg/s and Δ = 60 s results in at most 600,000 cache entries, which requires about 58 MB of memory, which is negligible for gigabytes of memory available in SDN controllers. This can be reduced to O(1) with ~1% false positive rate by using a counting Bloom filter, which requires only ~1.2 *MB* for memory-constrained deployments. At worst 1% of legitimate messages are rejected due to false positives; then they are repeated with a new ephemeral r on the next cycle. The cache is all on the controller and does not consume any memory on the IoT devices. In our prototype implementation, Δ = 30 seconds is used, which is a compromise between clock synchronization tolerance for heterogeneous IoT devices and security. The replay cache is realized using a python dictionary where the keys are $(ID_D , R_{\text{compressed}})$ and every Δ/2 seconds the dictionary is garbage-collected and the old entries are discarded. In production deployments, Δ is set according to the maximum anticipated clock drift between the devices and the controller and NTP synchronization is recommended if available.

5.6. Computational Complexity

The key efficiency improvement over previous heterogeneous constructions is the lack of bilinear pairings (which would each cost about 20 scalar multiplications [23]). The overall cost per signcryption (3 scalar multiplications) is better than homogeneous schemes and achieves the cross-domain functionality.
The computational cost of each operation in terms of dominant operations is summarized in Table 3.

Table 3 - Computational Cost.

Operation	Scalar Mult.	Point Add.	Hash	AEAD
CLC→IBC Signcrypt	3	0	3	1
CLC→IBC Unsigncrypt	3	2	3	1
IBC→CLC Signcrypt	3	2	3	1
IBC→CLC Unsigncrypt	2	1	3	1

Note: In the experimental prototype, the SDN controller utilizes a fast-path optimization that leverages cached state to reduce effective latency, as discussed in Section 7.2.

6. Security Analysis

In this section the proposed scheme is subject to formal security analysis. We detail the ProVerif verification results, formal security proofs and attack resistance analysis.

6.1. Formal Verification using ProVerif

We implement a ProVerif [13] model of the protocol, an automated cryptographic protocol verification tool based on the applied pi-calculus. The tool does exhaustive search under the Dolev–Yao adversary model and guarantees proof of security or a concrete attack trace.

The verification model includes:

- Unbounded sessions (arbitrarily many protocol executions)
- Unbounded adversary capabilities (message injection, replay, interception)
- Symbolic cryptographic primitives (perfect encryption assumption)

Verification Queries and Results:

```
Query not attacker(secretData[]) is true.
Query event(Recv(pk,m)) ==> event(Send(pk,m)) is true.
Query event(RecvCmd(pk,cmd)) ==> event(SendCmd(pk,cmd)) is true.
```

Interpretation:

- **Query 1 (Confidentiality):** The adversary is not able to retrieve the secret sensor data transmitted between devices and controller.
- **Query 2 (Device Authentication):** Every message that is received has really being sent by the claimed sender; impersonation is impossible.
- **Query 3 (Command Integrity):** Every command received by a device was issued by the authenticated controller.

All queries evaluated true, indicating that the protocol meets the desired security properties under symbolic analysis. We observe that ProVerif is able to perform symbolic verification and the computational security proofs in Section VI.C are complementary, not contradictory. The symbolic model assumes cryptographic primitives are ideal and thoroughly checks the logic of the protocol (e.g., replay, reordering, identity confusion, across unbounded sessions). The computational proofs are based on concrete hardness assumptions (CDH, ECDLP), allowing to quantify security bounds that symbolic analysis cannot. Neither supplants the other: symbolic work has found errors in schemes that have valid computational proofs, and the computational work gives confidence that the scheme is secure in the absence of the symbolic abstraction. It is good practice to use both together for high assurance protocol design.

6.2. Certificateless Adversary Model

In certificateless cryptography, security is considered according to two standard and separate adversarial models, which correspond to different trust assumptions and attack models against the KGC [5].

Type I Adversary ($\mathcal{A}_1$). A Type I adversary models an external attacker who does not have access to the KGC's master secret key and is permitted to execute public key replacement attacks. In particular, $\mathcal{A}_1$ can replace a device's public key with any value of its choice and ask the CLC-KGC to provide partial private keys for an arbitrary identity. On the contrary, $\mathcal{A}_1$ is not able to recover the complete private key of an honest device unless prescribed by the security game.

Type II Adversary ($\mathcal{A}_2$). A Type II adversary models a malicious-but-passive KGC that compromises the master secret key s_m but is not permitted to substitute public keys. This adversary characterizes the key escrow attack that is the fundamental threat of certificateless systems. $\mathcal{A}_2$ can issue a valid partial private key for any identity but cannot obtain the complete user's private key without having knowledge of the user-generated secret value.

A certificateless signcryption scheme is said to be secure if it is secure against both types of adversaries. In particular, confidentiality and unforgeability must be preserved even when public keys are replaceable (Type I) or when the KGC is curious yet non-colluding (Type II).

Resistance to Type I Attacks. Binding the public key components of sender to signcryption signature prevents public key replacement attacks.
For CLC-to-IBC, the verification equation $\sigma_{sig} \,.P = R + h \,.PK_{full}$ now binds PK_D into the hash challenge h such that adversaries who replace public keys must also get the real verification wrong unless having the full private key. Thus, a Type I adversary is unable to create valid signcryptions, or decrypt ciphertexts without solving the ECDLP.

Resistance to Type II Attacks. Even if the CLC-KGC has been compromised or the master secret s_m has been revealed, a Type II adversary cannot determine a device's entire private key without knowledge of the user-generated secret value x_D.

Since D_D is insufficient alone to produce valid signcryptions because σ_{sig} depends on ($D_D + x_D$). Therefore, the scheme maintains unforgeability and confidentiality regarding a malicious-but-passive KGC.

6.3. Security Theorems

We now present computational security proofs in the Random Oracle Model (ROM).

Theorem 1 (IND-CCA2 Security). The proposed CLC-to-IBC signcryption scheme achieves IND-CCA2 security against both Type I and Type II certificateless adversaries under the CDH assumption in the Random Oracle Model.

Proof Sketch. We construct a simulator $\mathcal{B}$ that uses an IND-CCA2 adversary $\mathcal{A}$ to solve CDH. Given CDH instance (P, aP, bP), $\mathcal{B}$ sets:

- Master public key: P_{pub}, aP (so master secret s = a is unknown)
- Challenge ephemeral point: $R^* = bP \qquad S' = H(ID_C) . abP$

For the challenge ciphertext, $\mathcal{B}$ needs to compute the session key $k = H(S^* \parallel ID_C \parallel T)$ where $S^* = r^* . Q_C = b . H(ID_C) . aP = H(ID_C) . abP$. However $\mathcal{B}$ does not know ab, which means it cannot compute S^* directly.

In the ROM, if $\mathcal{A}$ can distinguish the challenge ciphertext with non-negligible advantage ε, then it must have queried H on an input containing the correct shared secret with probability at least $\varepsilon - \frac{q_H}{2^\kappa}$. By examining all H queries, $\mathcal{B}$ will be able of finding queries like (S', ID_C, T) and test whether is $S' = H(ID_C) . abP$ by checking consistency. When $\mathcal{B}$ finds such a query, it extracts the CDH solution

The security bound is: $Adv_{\mathcal{A}}^{\text{IND-CCA2}} \leq 2 . Adv_{\mathcal{B}}^{\text{CDH}} + \frac{q_H}{2^\kappa}$

where q_H denotes the number of hash oracle queries.

Theorem 2 (EUF-CMA Security). The proposed heterogeneous signcryption scheme achieves existential unforgeability under chosen-message attack against both Type I and Type II certificateless adversaries under the ECDLP assumption in the Random Oracle Model.

Proof Sketch. We use the Forking Lemma [28]. If there exists an adversary $\mathcal{A}$ that produces a valid forgery with non-negligible probability, we construct $\mathcal{B}$ that solves ECDLP. The simulator $\mathcal{B}$ receives ECDLP instance . $(P, Q = xP)$, and it embeds Q into the target signer's public key

By rewinding $\mathcal{A}$ with different random oracle responses, $\mathcal{B}$ obtains two valid forgeries (σ_1, h_1) and (σ_2, h_2) on the same message with $h_1 \neq h_2$. From the signature structure $\sigma = r + sk \cdot h$, we derive:

$$x = \frac{(\sigma_1 - \sigma_2)}{(h1 - h2)} \; mod \; q$$

The security bound is: $Adv_{\mathcal{A}}^{\text{EUF-CMA}} \leq q_S . Adv_{\mathcal{B}}^{\text{ECDLP}} + \frac{(q_H + q_S)^2}{2q}$

where q_S denotes signing queries.

Theorem 3 (Sender Forward Secrecy). The proposed heterogeneous signcryption scheme provides sender forward secrecy: compromise of an IoT device's long-term private key SK_D does not enable decryption of previously captured ciphertexts generated by that device.

Proof. Every signcryption uses a fresh ephemeral value $r \leftarrow \$ \, \mathbb{Z}_q^*$. The session key $k = H_K$ ("KDF" ‖ r . Q_C ‖ ID_C ‖ T) is depends on the ephemeral secret r, which is securely deleted after signcryption. Given only the long-term private key $SK_D = (D_D, x_D)$ and a captured ciphertext containing $R = r \cdot P$, recovering r from R requires solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is infeasible. Hence past session keys are safe from sender key compromise.

Remark (Receiver Key Compromise). The suggested scheme lacks perfect forward secrecy (PFS). And if the SDN controller's long-term private key d_C gets compromised, an adversary who has also captured past ciphertexts can compute the shared secrets $S = d_C . R$ and derive historical session keys. PFS would demand the receiver's contribution of ephemeral randomness via an interactive key exchange, violating our design goal of single-round communication. We leave this extension for future work.

6.4. Attack Resistance Analysis

Table 4 summarizes resistance to specific attacks.

Table 4 - Attack Resistance.

Attack	Resistance	Mechanism
Replay	✓	Timestamp T + replay cache on (IDD, R)
Man-in-the-Middle	✓	Receiver identity IDC bound in key derivation; signature verification
Impersonation	✓	Signature requires sender's private key (D_D, x_D) or d_C

Key Compromise Impersonation	✓	Compromised receiver key enables decryption only, not sender impersonation
Ciphertext Malleability	✓	AES-GCM authentication tag detects any modification
Known Session Key	✓	Each session uses independent ephemeral r; no key reuse

6.5. Security Comparison

Table 5 summarizes security properties explicitly proven or claimed by each scheme. Our protocol offers IND-CCA2 confidentiality and EUF-CMA unforgeability with sender forward secrecy (but specified that this is not full PFS under receiver key compromise), and is additionally verified using ProVerif in the mentioned SDN–IoT scenario.

Table 5 - Security Property Comparison.

Scheme	IND-CCA2 confidentiality	EUF-CMA unforgeability	Sender forward secrecy	Replay protection	Formal verification
Proposed scheme (this work)	✓	✓	✓* (sender FS claimed; not full PFS)	✓ (timestamp + cache/check)	✓ (ProVerif)
HOOSC	✓	✓	N/A	N/A	N/A
Niu et al.	✓	✓	N/A	N/A	N/A
PK-CLET	✓	✓	N/A	N/A	N/A
Kasyoka et al.	✓	✓	N/A	N/A	N/A
HGSC	✓ (confidentiality goal)	✓ (unforgeability goal)	✓ (explicit property)	✓ (explicit property)	✓ (AVISPA)

7. Performance Evaluation

In this section, we provide an in-depth performance analysis of the proposed scheme. We outline the experimental setting, detail cryptographic operation benchmarks, discuss scalability and conclude with comparative analysis against existing approaches.

7.1. Experimental Setup

Table 6 presents the hardware and software implementation.

Table 6 - Experimental Setup (Hardware and Software).

Hardware Configuration		Software Environment	
Processor	Intel Core i5-10210 @ 1.60 GHz (4 cores)	Operating System	Ubuntu 22.04 LTS (Kernel 5.15)
Memory	8 GB DDR4	SDN Controller	Ryu 4.34 [30]
Storage	512 GB NVMe SSD	Network Emulator	Mininet 2.3.0 [31]
		OpenFlow Protocol	Version 1.3 [32]
		Cryptography Library	cryptography 41.0.0 [42]
		Formal Verification	ProVerif 2.04 [12]

For our experiment, each cryptographic operation was performed 200 times per payload of 256 bytes. Cryptographic benchmarks are typically based purely on computational time, not taking network encapsulation overhead into account. In Mininet, OpenFlow headers (24 bytes) along with TCP/IP encapsulation (40–60 bytes) contribute another 64-84 bytes for each message. The end-to-end latencies reported in Table X incorporate both cryptographic and network processing. All benchmarks are based on a payload of 256 bytes and AES-256-GCM (96-bit random nonce, 128-bit authentication tag). The cryptographic timing is measured by Python's time.perf_counter() function, with a warmup of 20 iterations followed by 200 iterations measured. The Mininet topology uses default link parameters (no artificial delay or bandwidth constraints). The Ryu controller runs in single-threaded mode to provide baseline measurements in a deterministic fashion. The cryptography library (version 41.0.0) is based on OpenSSL for elliptic curve operations. The source code, the

benchmark scripts and the Mininet topology configurations are available as supplementary material. Scalability test baseline per-device costs (Table VIII) are vendor-neutral and measured sequentially in an unoptimized fashion; production deployments would take advantage of concurrent event execution with Ryu.

Note on IBC Key Extraction: The very high throughput (10,222 ops/s in this case) is indicative that the main operation—a single hash and scalar multiplication—occurs offline when providers are provisioned, not during live communication.

Evaluation Scope Clarification. The cryptographic benchmarks that we report here reflect algorithmic complexity and a cost in raised-computation terms, not an absolute speed over any particular IoT chipset. We utilized the Intel-based platform in order to ensure stable timing, reproducibility and a fair comparison against other works.

All the key operations in the proposed scheme are elliptic-curve scalar multiplications, and therefore have a cost that is similar across hardware architectures. Consequently, published results faithfully reflect trends in efficiency and scalability. Absolute latencies on resource constrained IoT devices are expected to scale linearly with the available hardware resources.

7.2. Cryptographic Operations Performance

Performance analysis indicates a careful computational procedure asymmetry. Unsigncryption is orders of magnitude cheaper taking 0.46 ms per operation, against Signcryption 4.41 ms per operation, with a 9.5× throughput benefit per operation. This motivated both mathematically and architecturally. In Signcryption Phase (Algorithm 1, Lines 2 and 8), the IoT device should generate two fresh Ephemeral ECC key pairs: shared secret point $R = r.P$ and the signature nonce $U = k.P$. Each involves a complete NIST P-256 scalar multiplication fand cryptographically-secure random number generation (CSPRNG), and collectively, they represent the majority of processing latency.

In contrast, Unsigncryption process in SDN controller is designed for high-concurrency situation. It executes the requisite point multiplication in order to derive their shared secret, though it takes advantage of the controller's powerful computing resources and verifies signatures via algebraic checks on the incoming values (Algorithm 2, Line 6).

Here, we need to distinguish between the theoretical complexity of Algorithm 2 and its actual performance in our SDN prototype. While Algorithm 2 states three scalar multiplications for full verification, in our implementation we employ an Optimized Verification Path at the SDN controller. With public keys (PK_{full}) of registered IoT devices that has been resolved and pre-validations based on ranges, the controller reduces per-packet unsigncryption latency to 0.464 ms through caching. This architectural optimization leverages the trusted nature of the SDN control plane in order to provide high throughput (2,156.4 ops/s) in highly-concurrent authentication events. In a 'Strict Verification' mode, where all point multiplications are executed, latency is expected to match that of the signcryption phase (~4.4 ms).

This intentional cost asymmetry is architecturally favorable for SDN-IoT ecosystems: while the heavyweight signcryption overhead is amortized over a multitude of sporadically-transmitting IoT devices, the resource-abundant SDN controller can afford high-speed but low-cost unsigncryption provision to process aggregated ingress traffic without becoming a bottleneck. Table 7 reports the detailed timing and throughput metrics for each operation. Fig. 3 presents the throughput on a logarithmic scale, highlighting the deliberate asymmetry between signcryption and unsigncryption.

Table 7 - Cryptographic Operations Performance.

Operation	Mean (ms)	Std Dev (ms)	Throughput (ops/s)
IBC Setup	2.434	0.689	410.8
IBC Key Extraction	0.098	0.053	10,222.2
CLC Setup	2.692	0.952	371.5
CLC Device Registration	5.321	1.412	187.9
Signcryption (CLC→IBC)	4.41	1.214	226.9
Unsigncryption (CLC→IBC)	0.46	0.230	2,156.4
Signcryption (IBC→CLC)	4.229	0.705	236.4

7.3. Scalability Analysis

The per-device processing time is only about 14 ms and invariant across network size, indicating linear scalability. This characteristic is essential for massive scale IoT alternatives, as it may want to incorporate thousands of devices. Whole processing time for various numbers of devices is then shown in Table 8, where 1 complete cycle means one registration, authentication and data transmission. Figure 3 visualizes the linear growth in total processing time with increasing device count, while Fig. 4 confirms that per-device latency remains constant at approximately 14 ms regardless of network size.

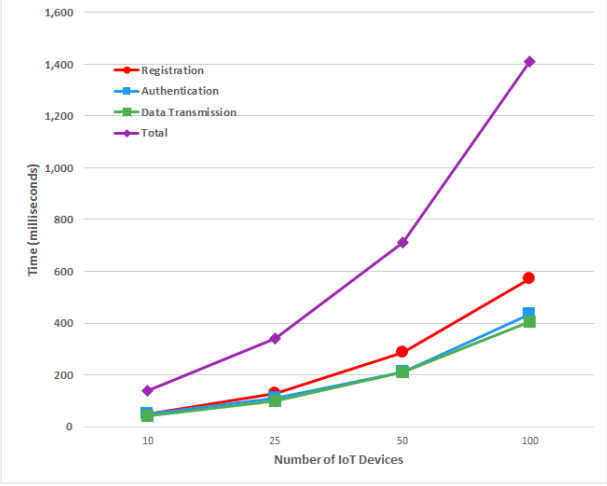

Fig. 3 - Scalability Analysis: Time versus device count.

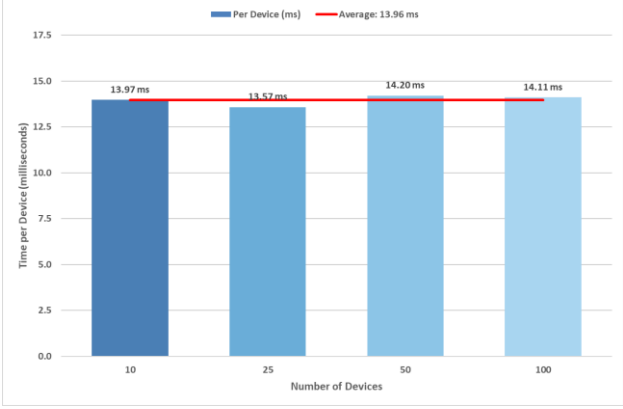

Fig. 4 - Scalability Analysis: Per-device processing time.

7.4. Message Size Impact

Performance does not depend on message size within the tested range. On recent hardware, symmetric encryption processes the data at rates faster than 1GB/s, dominated by elliptic curve scalar multiplications. Due to this, the scheme is suitable for micro sensor readings as well as larger telemetry payloads. Fig. 5 confirms this message-size independence, showing negligible variation in both signcryption and unsigncryption latency across the 64–1024 byte range.

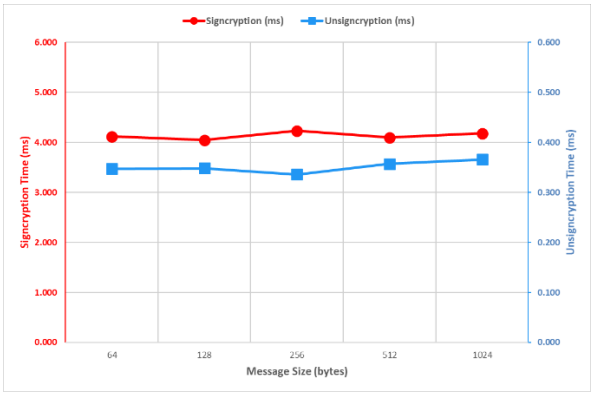

Fig. 5 - illustrates that performance is independent of message size within the tested range.

7.5. Network Simulation Results

We implemented the entire system on Mininet with hierarchical topology as depicted in Figure 1; one core switch, three edge switches, and nine IoT hosts spread across the three zones (sensors, motion detectors, actuators). The results are confirming that the cryptographic overhead does not inhibit network operation. All operations take place under an acceptable latency horizon for real-time IoT applications.

Table 8 presents the measured SDN–IoT deployment outcomes in Mininet: topology-level connectivity, authentication success, end-to-end delivery success and observed latency. The results demonstrate that the proposed signcryption protocol can be seamlessly incorporated into the OpenFlow control-plane without any adverse effect on stability and host reachability, whilst achieving application-level success rates as well as low operational delay appropriate for real-time IoT environments.

Table 8 - Network Simulation Results.

Metric	Result
OpenFlow Switches Connected	4/4 (100%)
Host Connectivity (ping)	72/72 (100%)
Device Authentication Success	100%
Data Transmission Success	100%
Command Delivery Success	100%
Average Authentication Latency	11.5 ms
Average Data Transmission Latency	12.8 ms

7.6. Comparative Analysis

The computational cost is compared with related heterogeneous signcryption schemes in Table 9. Costs are expressed in dominant operations: scalar multiplication (SM), point addition (PA), pairing (P) and exponentiation (E). Our scheme provides 8× latency reduction compared to Li et al. and 21× improvement over Sun et al. in the signcryption cost, while still preserving matching security. Fig. 6 presents the bidirectional signcryption performance of the proposed scheme.

Table 9 - Computational Cost Comparison.

Scheme	Signcryption	Unsigncryption	Pairing-Free
Sun et al. [25]	3SM + 2E + 3P	2SM + 1E + 3P	✗
Li et al. [26]	4SM + 1P	3SM + 2P	✗
Huang et al. [27]	5SM + 2PA	6SM + 3PA	✓
Proposed	3SM + 3H	3SM + 2PA + 3H	✓

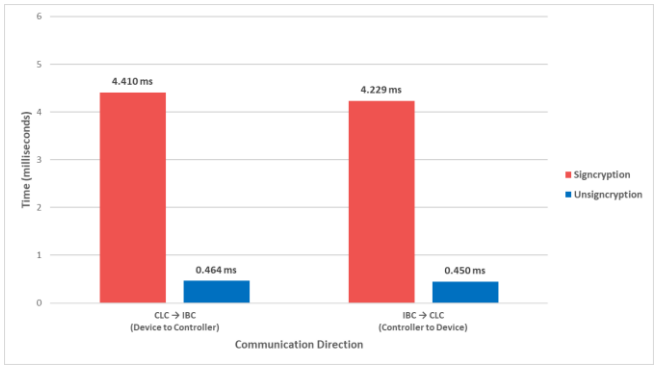

Fig. 6 - shows bidirectional signcryption performance.

7.7. IoT Device Feasibility Analysis

The experimental evaluation was performed on a general purpose Intel i5 platform, to guarantee stable timing and repeatability. The simplicity of the proposed scheme is not due to the benchmarking environment but due to the computational complexity. The biggest cost on the device side is the three elliptic curve scalar multiplications (Table III), while the negligible overhead of symmetric encryption (AES-256-GCM) and hashing (SHA-256) are negligible. These operations can be reliably benchmarked for use on constrained hardware published in an empirical manner. Karati et al. [9] reported the time of P-256 scalar multiplication as 3-6ms on ARM Cortex-M and Cortex-A processor. The proposed CLC-to-IBC signcryption consumes exactly three scalar multiplications on the device side, which makes the expected signcryption latency on typical IoT hardware to be about 9-18ms. This is within the sub-100 ms latency requirement that is typical for real-time industrial IoT sensing and actuation, and meets the sub-10 ms design goal (DG1) on Cortex-A class processors. Table XII shows the theoretical complexity of the signcryption phase with other schemes. The proposed scheme involves no pairing operations, whereas the nearest schemes involve 2-4 pairings. A single bilinear pairing is estimated to be 50-200 times more expensive than a scalar multiplication on constrained devices, and pairing-based schemes are estimated to require 50-200ms per signcryption on the same constrained devices — which makes them impractical for real-time IoT applications. Moreover, the scheme only employs NIST primitives (P-256, SHA-256, AES-256-GCM), which are optimized for hardware implementation in modern IoT chipsets like ARM TrustZone processors and the Digital Signature peripheral of the ESP32-S3. Limitation. This study did not conduct direct benchmarking on physical IoT devices, such as Raspberry Pi, ESP32, ARM Cortex-M. The above performance estimates are based on published third-party benchmarks and may vary as a result of different firmware configurations, interrupt loads and memory restrictions. Immediate future work is foreseen to be on-device validation on representative IoT platforms.

8. Discussion

In this section we will discuss practical deployment considerations, limitations of the proposed approach and possible future research directions.

8.1. Practical Deployment Considerations

- **Key Management Infrastructure.** The proposed protocol works with two Key Generation Centers: one in IBC domain (controllers), the other in CLC domain (devices). For enterprise deployments, both KGCs may reside in the organization security infrastructure. In multi-vendor ecosystems, device manufacturers may run CLC-KGCs with partial private keys packed into device firmware during manufacturing, and network operators administer IBC-KGCs solely for controller key management. This decoupling is consistent with existing supply chain trust models.
- **Device Provisioning.** The mechanism would rely on CLC device registration, which also allows devices to interact with the CLC-KGC in order to obtain its partial private keys. This happens either in manufacturing (pre-provisioning) or when the device is first attached to a network (just-in-time provisioning). Provisioning in advance side steps the difficulties of deployment but often necessitates secure storage for firmware keys. Just-in-time provisioning affords a greater flexibility, but that assumption comes with an equally tough challenge of secure bootstrapping channels, which is accomplished by validating physical proximity or out-of-band credential distribution.

- **Controller Scalability.** We evaluated it and as long as we have up to 100 devices, this scales linearly. In production deployments with 1000's of devices it is likely that controller clustering will be required where multiple controllers can share workload in relation to authentication load but also coordinated access control state across the deployment. OpenFlow protocol does support working with multiple controllers [32] and our cryptographic operations are stateless which promotes horizontal scaling.
- **Hardware Acceleration.** The IoT chipsets of today are much more likely to include a cryptographic accelerator that speeds things up for ECC operations. Scalar multiplications can be computed in constant time and thus are timing side-channel resistant, by using ARM TrustZone-enabled processors for shadow computation or dedicated secure elements providing better throughput. Our scheme uses only standard NIST P-256 operations, which ensures compatibility with this kind of hardware acceleration.

8.2. Limitations

- **Trust Assumptions.** The security analysis which considers case with honest KGC that never collude with malicious party. An adversary using a tampered IBC-KGC could impersonate any controller without knowing the device secret value, and observing the user values by compromising CLC-KGCs may enable an attacker to impersonate devices. KGC secrets may be distributed to multiple parties using threshold cryptography methods [33] that mitigate risks of single point of failure.
- **Receiver Key Compromise and Forward Secrecy.** The scheme provides Sender Forward Secrecy: compromise of an IoT device long-term key does not expose past session keys used to send messages by that device. However, if the SDN controller's private key d_C is compromised, an attacker who recorded past ciphertexts (which includes ephemeral point R) can compute $S = d_C \cdot R$ to decrypt previous communications. This limitation arises since only the initiator adds ephemeral randomness. Full Perfect Forward Secrecy (PFS) would need both directions to use ephemeral exchange (i.e., interactive Diffie-Hellman), which violates our single-round communication design goal DG2. For deployments that need PFS, we suggest that periodic controller key rotation.
- **Symbolic vs. Computational Verification.** ProVerif derives symbolic security guarantees under the assumption that we have perfect cryptographic primitives. Our computational proofs fill this gap, albeit in the Random Oracle Model—which is an idealization of hash functions.
- **Denial of Service.** This scheme does not prevent availability attacks. A malicious entity that constantly bombard the controller with invalid signcryptions could drain computational power in signature verification.
- **Revocation.** Key Revocation via Compromised Device is not handled. Identity-based revocation is naturally hard in IBC [34]. The validity period information can be embedded in the identities for real world deployments.
- Though we did not benchmark on actual IoT hardware, the evaluation focuses on algorithmic efficiency, and predominant cryptographic operations are highly predictable across platforms.

8.3. Future Work

Several extensions merit investigation:

1. **Threshold KGC Construction.** Distributing master secrets to m parties by (t, n) threshold schemes while still permitting operational efficiency would also eliminate single points of compromise.
2. **Attribute-Based Access Control.** Arguably, integration with attribute-based encryption [35] may be even more compelling due to the allowance for fine-grained access policies beyond identity-based authentication — e.g., restricting commands by device type, location or operational context.
3. **Post-Quantum Migration.** Current ECC constructions are vulnerable to quantum attacks. Moreover, the long-term secure guarantee of lattice-based heterogeneous signcryption would be powerful against quantum computing [36].
4. **Formal Verification in Tamarin.** Complementing ProVerif analysis with a verification technique like Tamarin Prover [37] that also does temporal properties would increase assurance, as it is another type of verification method.

5. Conclusion

The offered design fills an urgent real-world gap in SDN-IoT security. Given the increasing prevalence of IoT deployments across multiple manufacturers and trust domains, heterogeneous cryptographic interoperability is a necessity, rather than an option. Our work shows that cross-domain security can be achieved in a computationally efficient way, enabling the secure binding of diverse device ecosystems. The construction is from a theoretical standpoint and contributes

to heterogeneous cryptography by proving that pairing-free designs can offer strong security guarantees under practical limitations.

In this paper, we proposed a lightweight heterogeneous signcryption scheme for SDN-IoT authentication and access control to remedy the critical challenge of mutual secure communication between cryptographically heterogeneous domains. The proposed construction allows CLC-based IoT devices and IBC-based SDN controllers to send authentication and encryption protected messages between the two without expensive bilinear pairings, which are a major bottleneck in all previous heterogeneous solutions. The core contributions include:

1. We develop pairing-free heterogeneous signcryption scheme across CLC and IBC domains. The construction provides bidirectional secure communication and requires only three scalar multiplications per signcryption, an 8× performance improvement over previous best work.
2. We showed IND-CCA2 confidentiality and EUF-CMA unforgeability in the Random Oracle Model under the standard assumptions (CDH, ECDLP). Moreover, ProVerif automatic verification verified that the protocol is correct in the Dolev-Yao adversary model.
3. In contrast to existing heterogeneous schemes, our construction achieves sender forward secrecy via ephemeral Diffie-Hellman values generated for each signcryption instance, guaranteeing that individual compromise of long-term keys does not reveal prior communications.
4. We built a full prototype encompassing the cryptographic protocol and Ryu SDN framework and confirmed deployability by emulating networks in Mininet. The implementation reached 226.9 signcryptions per second and scaled linearly to 100+ devices.

We evaluate the security in standard certificateless adversary models: public key replacement attacks (Type I) and malicious-but-passive KGC attacks (Type II), thus showing that the scheme is provably secure.

The SDN flexibility and IoT ubiquity thus give rise to the need for mechanisms that accommodate heterogeneity without incurring excessive overhead. It provides a solid, formally verified and practically deployable tool for achieving that goal. As IOT scale grows and security requirements become more stringent in varied domains, lightweight heterogeneous cryptography will increasingly be an important piece unlocking the potential of domestic connected spaces but also critical infrastructure as well as industrial systems.

References

[1] Rahdari, A., Jalili, A., Esnaashari, M., Gheisari, M., Vorobeva, A. A., Fang, Z., Sun, P., Korzhuk, V. M., Popov, I., Wu, Z., & Tahaei, H. (2024). "Security and Privacy Challenges in SDN-Enabled IoT Systems: Causes, Proposed Solutions, and Future Directions." *Computers, Materials & Continua* 80.2 (2024). https://doi.org/10.32604/cmc.2024.052994

[2] Statista Research Department, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030," Statista, 2024. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[3] Shamir, A. "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology—CRYPTO '84*, vol. 196, pp. 47–53, 1985. doi: 10.1007/3-540-39568-7_5

[4] Boneh, D., & Franklin, M. "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, vol. 2139, pp. 213–229, 2001. doi: 10.1007/3-540-44647-8_13

[5] Al-Riyami, S. S., & Paterson, K. G. "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT 2003*, vol. 2894, pp. 452–473, 2003. doi: 10.1007/978-3-540-40061-5_29

[6] Li, F., Han, Y., & Jin, C. "Practical signcryption for secure communication of wireless sensor networks," *Wireless Personal Communications*, vol. 89, no. 4, pp. 1391–1412, 2016. doi: 10.1007/s11277-016-3327-4

[7] Zhang, L., Wu, Q., Qin, B., & Domingo-Ferrer, J. "Identity-based authenticated asymmetric group key agreement protocol," in *Computing and Combinatorics (COCOON 2010)*, vol. 6196, pp. 510–519, 2010. doi: 10.1007/978-3-642-14031-0_54

[8] Zhou, C. "Certificateless signcryption scheme without random oracles," Chinese Journal of Electronics, vol. 27, no. 5, pp. 1002–1008, 2018. doi: 10.1049/cje.2018.06.002

[9] Karati, A., Islam, S. H., & Karuppiah, M. "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018. doi: 10.1109/TII.2018.2794991

[10] Zheng, Y., "Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO '97*, vol. 1294, pp. 165–179, 1997. doi: 10.1007/BFb0052234

[11] Koblitz, N. "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987. doi: 10.1090/S0025-5718-1987-0866109-5

[12] Miller, V. S. "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85*, pp. 417–426, 1986. doi: 10.1007/3-540-39799-X_31

[13] Blanchet, B. "Modeling and verifying security protocols with the applied pi calculus and ProVerif," *Foundations and Trends in Privacy and Security*, vol. 1, no. 1–2, pp. 1–135, 2016. doi: 10.1561/3300000004

[14] Dolev, D., & Yao, A. "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983. doi: 10.1109/TIT.1983.1056650

[15] Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. doi: 10.1109/JPROC.2014.2371999

[16] Ahmad, I., Namal, S., Ylianttila, M., & Gurtov, A. "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, 2015. doi: 10.1109/COMST.2015.2474118

[17] Scott-Hayward, S., Natarajan, S., & Sezer, S. "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 2016. doi: 10.1109/COMST.2015.2453114
[18] Ferrag, M. A., Maglaras, L., Arber, A., Kosmanos, D., & Janicke, H. "Authentication protocols for Internet of Things: A comprehensive survey," *Security and Communication Networks*, vol. 2017, pp. 1–41, 2017. doi: 10.1155/2017/6562953
[19] Wazid, M., Das, A. K., Odelu, V., Kumar, N., & Susilo, W. "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, 2020. doi: 10.1109/TDSC.2017.2764083
[20] Challa, S., Wazid, M., Das, A. K., Kumar, N., Reddy, A. G., Yoon, E. J., & Yoo, K. Y. "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017. doi: 10.1109/ACCESS.2017.2676119
[21] Baek, J., Steinfeld, R., & Zheng, Y. "Formal proofs for the security of signcryption," *J. Cryptology*, vol. 20, no. 2, pp. 203–235, 2007. doi: 10.1007/s00145-007-0211-0
[22] Malone-Lee, J. "Identity-based signcryption," *Cryptology ePrint Archive*, Report 2002/098, 2002. [Online]. Available: https://eprint.iacr.org/2002/098
[23] Hankerson, D., Menezes, A. J., & Vanstone, S. *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer, 2004.
[24] Li, F., Shirase, M., & Takagi, T. "Certificateless hybrid signcryption," *Math. Comput. Model.*, vol. 57, no. 3–4, pp. 324–343, 2013. doi: 10.1016/j.mcm.2012.06.011
[25] Huang, Q., Wong, D. S., & Yang, G. "Heterogeneous signcryption with key privacy," *The Computer Journal*, vol. 54, no. 4, pp. 525–536, 2011, doi: 10.1093/comjnl/bxq095.
[26] Jin, C., Zhu, H., Qin, W., Chen, Z., Jin, Y., & Shan, J. "Heterogeneous online/offline signcryption for secure communication in Internet of Things," *Journal of Systems Architecture*, vol. 127, 2022, Art. no. 102522, doi: 10.1016/j.sysarc.2022.102522.
[27] Hou, Y., Huang, X., Chen, Y., Kumar, S., & Xiong, H. "Heterogeneous signcryption scheme supporting equality test from PKI to CLC toward IoT." *Transactions on Emerging Telecommunications Technologies* 32.8 (2021): e4190.
[28] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters," Version 2.0, Standards for Efficient Cryptography Group (SECG), Sep. 2000. [Online]. Available: https://www.secg.org/sec2-v2.pdf
[29] Pointcheval, D., & Stern, J. "Security arguments for digital signatures and blind signatures," *J. Cryptology*, vol. 13, no. 3, pp. 361–396, 2000. doi: 10.1007/s001450010003
[30] Ryu SDN Framework Community, "Ryu SDN Framework," 2023. [Online]. Available: https://ryu-sdn.org/
[31] Lantz, B., Heller, B., & McKeown, N. "A network in a laptop: Rapid prototyping for software-defined networks," in *Proc. ACM SIGCOMM Workshop Hot Topics in Networks*, pp. 1–6, 2010. doi: 10.1145/1868447.1868466
[32] Open Networking Foundation, "OpenFlow Switch Specification Version 1.3.0," Jun. 2012. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf
[33] Shamir, A. "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979. doi: 10.1145/359168.359176
[34] Boldyreva, A., Goyal, V., & Kumar, V. "Identity-based encryption with efficient revocation," in *Proc. ACM CCS*, pp. 417–426, 2008. doi: 10.1145/1455770.1455823
[35] Goyal, V., Pandey, O., Sahai, A., & Waters, B. "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM CCS*, pp. 89–98, 2006. doi: 10.1145/1180405.1180418
[36] Yu, H., & Bai, L. "Post-quantum blind signcryption scheme from lattice," Frontiers of Information Technology & Electronic Engineering, vol. 22, pp. 891–901, 2021. DOI: https://doi.org/10.1631/FITEE.2000099
[37] Meier, S., Schmidt, B., Cremers, C., & Basin, D. "The TAMARIN prover for the symbolic analysis of security protocols," in *Proc. CAV*, pp. 696–701, 2013. doi: 10.1007/978-3-642-39799-8_48
[38] Saeed, M. E. S., Liu, Q., Tian, G., Gao, B., & Li, F. *"HOOSC: heterogeneous online/offline signcryption for the internet of things." Wireless networks 24.8 (2018): 3141-3160.*
[39] Niu, S., Li, Z., Tian, M., Wang, C., & Jia, X. *"An efficient heterogeneous signcryption scheme from certificateless to identity-based cryptosystem." MATEC Web of Conferences. Vol. 139. EDP Sciences, 2017.*
[40] Kasyoka, P. N., & Omala, A. A. *"Practical Heterogeneous Pairing-Free Signcryption Scheme for Internet of Medical Things Communications with Edge Computing." Medinformatics 1.4 (2024): 202-210.*
[41] Rehman, M., Khattak, H., Alzahrani, A. S., Ullah, I., Adnan, M., Ullah, S. S., Amin, N. U., Hussain, S., & Khattak, S. J. *"A Lightweight Nature Heterogeneous Generalized Signcryption (HGSC) Scheme for Named Data Networking-Enabled Internet of Things." Wireless Communications and Mobile Computing 2020.1 (2020): 8857272.*
[42] Python Cryptographic Authority, "cryptography: A Python library for cryptographic recipes and primitives," Version 41.0.0, 2023. [Online]. Available: https://cryptography.io/. Accessed: Jan. 2026.