RESEARCH ARTICLE

A Conditional Privacy-Preserving Short Signature Scheme for Industrial Internet of Things

Pushpendra Kumar Vashishtha[1,2,*] **and Saru Kumari**[2]

[1] Department of Mathematics, D.A.V. (P.G.) College, Muzaffarnagar, Affiliated to Maa Shakumbhari University, Uttar Pradesh 251001, India
[2] Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India

Abstract

Securely transmitting a large amount of data in a short time is a serious challenge in today's digital age. Cryptographic primitives can significantly alleviate this problem. The invention of digital signatures represents a major advance in this area. The Certificateless Aggregate Signature (CLAS) scheme is a cryptographic primitive that greatly reduces the computation cost by aggregating several signatures into a single short signature. However, the costs of aggregate signatures have not been reduced to the desired extent. In this paper, we propose a certificateless aggregate signature scheme that requires only two bilinear pairing operations to verify both a single signature and an aggregate signature, regardless of the number of signers. This makes the scheme highly suitable for low-cost IoT applications. In this scheme, we also present a Type I attack on Horng et al.'s scheme and provide an improved version.

Keywords: certificateless, aggregate signature, low cost, Type I attack, conditional privacy.

Submitted: 01 January 2026
Accepted: 23 March 2026
Published: 24 March 2026

Vol. 2, **No.** 1, 2026.
10.62762/JRSC.2026.376190

***Corresponding author:**
✉ Pushpendra Kumar Vashishtha
pktha.iitkgp@gmail.com

1 Introduction

In today's fast-paced world, the IoT has made life much easier and more convenient. The Internet of Things permeates every aspect of our lives, be it the medical and healthcare sector, transportation, or agriculture [1]. People use IoT everywhere. IoT applications transmit and share daily data, but the security of such vast data remains a major challenge. To address these challenges, many authors have contributed in various ways over time. First, Diffie and Hellman [2] discovered public key cryptography, which ushered in a new era. Public key cryptography uses a trusted authority to certify a user's identity. It provides a certificate based on that identity, and this certificate is used to verify the user's identity globally. However, a problem arose, known as the certificate management problem.

There should be no need for any third-party certificate authority to authenticate the identity of the users. Keeping this goal in mind, Shamir [3] invented a new paradigm, which he named Identity-Based Signature scheme. Within this paradigm, the user's public key authenticates the validity of their identity rather than relying on a certificate. Shamir derived the public key of the user from his identity, and a third party named Private Key Generator would generate the private

Citation
Vashishtha, P. K., & Kumari, S. (2026). A Conditional Privacy-Preserving Short Signature Scheme for Industrial Internet of Things. *Journal of Reliable and Secure Computing*, 2(1), 39–49.

key of each user and then provide it to each user. If the PKG gets all users' private keys, it can crash the system. Therefore, this type of problem is named as the escrowing the key problem [4].

Al-Riyami et al. [5] laid the foundation for a primitive that verifies users' identities without certificates and prevents a trusted third party from having full control over the user's private keys. They called this certificateless cryptography. While a third party exists, it only contributes a small portion of the user's private key. The third party's function is to give a partial secret key to the user, thereafter, the user selects a random number and amalgamates this partial secret key with another secret key to produce a novel private key. This way, there's no need for a certificate to verify users' identities, nor does the third party possess the user's private key. This method eliminates both of the issues that have caused significant frustration for a long time.

Many applications, such as VANETs [6–9], sensor networks [10], industrial internet of things [11, 12], agriculture, and e-voting, require verifying large amounts of data against their signatures in real time. The time required to verify n signatures is n times the time required to verify a single signature. However, Boneh et al. [13] developed a unique primitive that allows verifying a single signature instead of multiple signatures, saving significant time. Combining certificateless signatures with this new approach (called Aggregate Signature) creates a new paradigm known as the Certificateless Aggregate Signature Scheme.

When the Internet of Things (IoT) is deployed within industrial settings, it is referred to as the Industrial Internet of Things (IIoT) [14]. IIoT is utilized in industries to collect real-time data, enhance manufacturing efficiency, and mitigate unexpected equipment failures. Through the application of IIoT, we can continuously monitor production processes and improve product quality in real time [15]. It also employed in data analytics to facilitate decision-making. IIoT systems are equipped with sensors that collect real-time data. Subsequently, actuators execute actions based on prevailing conditions; however, securely transmitting this data from one machine to another—while ensuring data integrity and authenticating the identity of the sending machine—requires specialized measures [16]. To achieve this, we employ cryptography. In this paper, we propose integrating a certificateless aggregate signature scheme primitive with the Industrial Internet of Things to safeguard machine identities and enable the rapid verification of data validity.

2 Related Work

Gentry et al. [17] combined the aggregate signature with identity based to give identity based aggregate signature. This leads to further formation of the aggregate signature schemes. Gong et al. [18] created two certificateless signature schemes with different characteristics. Huang et al. [19] classified the adversary as normal, strong and super as per their strength of having secret keys. Xiong et al. [20] proposed a verifiable signature technique that asserts computational efficiency, requiring only three pairing operations as the computational cost.

Horng et al. [21] proposed a scheme with the advantage of conditionally tracing vehicles in vehicular sensor networks and proved its security in the ROM under the hardness assumption of the CDHP. Later, Li et al. [10] demonstrated a Type II attack on this scheme in 2016. Therefore, to avoid redundancy, we include only the Type I attack and its subsequent improvements in the manuscript. The scheme in [21] requires three bilinear operations, whereas our scheme requires only two bilinear pairing operations. The aggregate signature scheme in [22] avoids bilinear operations to reduce power consumption and proves security in the ROM under the ECDLP assumption.

Kar et al. [23] proposed a new signature scheme for wireless sensor networks and claimed low power consumption. However, the scheme depends on n, the number of signers, and its computational cost increases linearly with n, making it inefficient for IoT application deployment. Recently, Vashishtha et al. [24] revealed weaknesses in this scheme by presenting two distinct types of attacks and proposed an improvement, but did not provide a formal security proof or performance evaluation. Later, Xiong et al. [25] proposed a collusion-resistant scheme, which was subsequently broken in [26].

The paper flows in the following way: Section 3 gives basic information required to understand the technicalities in the paper along with network model and security model. Section 4 provides Horng et al. [21] scheme in tabular form and then undergoes a Type I attack followed by the improvement over the scheme. Section 5 shows the security analysis of the scheme. Section 6 demonstrates the performance of our scheme with regard to CostS, CostV, and CostVAgg.

Section 7 gives the conclusion.

3 Preliminaries

This section provides the basic information that is necessary to understand the paper. The first term is CDHP, which we have used frequently in our paper. In this problem, for a given triplet $\{P, lP, mP\}$, which lie on the elliptic curve group G_1, it is not possible to calculate lmP in polynomial time. Second, we have used Type I and Type II adversaries in scheme. The first adversary is lucky because it gets the power to access the sensor node's public key and then replaces it with their public key with a value of its choosing. The second Type II adversary is a malevolent Key Generation Center (KGC) and hence possesses the power to get an access to the system's private key or can get the partial secret key $snpark_i$ of the sensor node SN_i.

The acronyms that are used throughout the paper are presented in Table 1 and the system model of CLAS scheme is shown in Table 2.

Table 1. Notations and abbreviations.

Notation	Description
l	Security parameter
KGC	Key generation center
e	Bilinear map
P, Q	Generators for the group G_1
$\{\alpha, P_{pkey}\}$	Secret key and public key of KGC
$\{\beta, TRA_{pkey}\}$	Secret key and public key of TRA
$Ha_{a0}, Ha_{a1}, Ha_{a2}$	Hash functions
$ReI_i, PseI_i$	Real and pseudo-identity of sensor node SN_i
$\{snpark_i, snsk_i, snpk_i\}$	Partial-private, secret and public key of SN_i
sig_i, m_i	Signature on message m_i
CostP	Bilinear pairing cost
CostM	Scalar multiplication cost
CostA	Point addition cost
{CostS, CostV, CostVAgg}	Signature creation, verification and aggregate verification cost.

3.1 Network Model

In our network model we have the following entities as shown in Figure 1. The description of each entity is as follows:

Sensor Node It receives the data from surrounds and creates a signature to the data and send the data to nearest Machine. We denote the ith sensor node as SN_i in our scheme.

Machine Machine collects the data with signatures from the sensor nodes and verifies the signatures and then send to Manager to aggregate the signatures. We denote the ith machine as M_i in our scheme.

Manager Manager verifies the aggregated signatures and take further action based on the data. Manager is equipped with high computational resources. We denote the ith enterprise manager as ME_i in our scheme.

TRA Trusted authority in our scheme is TRA. TRA is the administrator of the system and is responsible for generating the pseudo-identity of each sensor node SN_i and also responsible for removing the malicious sensor node by tracing the idenity of sensor node.

KGC This is an outsider private authority. It is not a part of IIoT network but it generates the partial secret key of each sensor node after taking the pseudo-identity of the sensor node. Once the partial secret key is granted to TRA, KGC goes offline.

In our scheme, TRA is the only trusted authority and KGC is semi trusted. Sensor nodes send their real identities to TRA and TRA generates pseudo-identity to each sensor node and then sends these pseudo-identities to KGC and sensor nodes. KGC then generates the partial secret key to each sensor node and sends back to TRA and then TRA sends it to sensor node.

3.2 Security Model

In our security model, we consider two types of adversaries: one is referred to as a Type I adversary, and the other as a Type II adversary. Our security model is a game-based interaction model in which a game is played between an adversary and a simulator. This game is query-based; the adversary submits queries to a challenger, and the simulator responds to them accordingly. Following this interaction, the simulator is required to solve a computationally

Table 2. A system model of CLAS scheme.

Algorithm	Input	Output
Setup	Security parameter l	Params, KGC keys, TRA keys
Pseudo-identity	Real identity, TRA secret key	Pseudo-identity
Partial secret key	Pseudo-identity, KGC secret key	Partial secret key
Sensor node key	Pseudo-identity	Sensor node keys
Sign	Message, Pseudo-identity	Signature
Verify	Signature, Public keys, Pseudo-identity	True or False
Aggregate	Sensor nodes' signatures	Aggregate signature
Aggregate verify	Aggregate signature, Public keys, Pseudo identities	True or False

hard problem. Thus, our security model effectively translates into a form of reduction security. We proceed by assuming that the adversary is capable of breaking the security of our scheme, and we then demonstrate that, if such a feat is possible, the simulator can successfully solve the underlying computationally hard problem.

Game I The game is played between a Type-1 adversary A_1 and a simulator S_1. The Type-1 adversary generates various queries to the simulator S_1 such as queries to create a sensor node namely $CUQuery$, generate a secret key $SKQuery$, replace a public key $RKQuery$ and retrieve a partial secret key $PSKQuery$ and the sign creation query $SQuery$ and the simulator responds to all these queries accordingly. Subsequently, the adversary presents a forged signature. This forged signature is then subjected to verification conditions; if it successfully passes these checks, we conclude that the adversary has breached security or successfully forged the signature. Our scheme is deemed secure only if no such adversary is able to produce a valid, existentially forged signature within polynomial time. The following queries are the part of Game I.

$CUQuery$: Adversary gives the pseudo-identity as input and the simulator sends back the corresponding public key.

$SKQuery$ Adversary raises the query with the input of pseudo-identity and gets back the corresponding secret key.

$PSKQuery$: Adversary submits this query with an input of pseudo-identity and gets back the corresponding partial secret key.

$RKQuery$: Adversary wants to replace the public key of any sensor node. This specific strength is provided to Type I adversary only. Here adversary wants to replace the public key with the value chosen by itself. For this purpose, adversary submits this query with the input of pseudo-identity and chosen public key (to be replaced), and the simulator replaces the public key of the desired sensor node.

$SQuery$: This query is raised to get the signature of message m_i and pseudo-identity $PseI_i$ and adversary A_1 gets back the valid signature. We say that adversary wins the Game I and is successful in forging the signature if the following conditions hold:

1. Adversary A_1 never makes the $PSKQuery$ for the target sensor node identity.
2. Adversary A_1 never makes the $SQuery$ for the target identity and target message.
3. Adversary A_1 successfully forges the signature.

Game II This game is identical to Game I except that the adversary A_2 has the power to get the partial secret key or the master key of KGC. This adversary can not replace the public key of any sensor node. The queries and response are same as in Game I.

We say that adversary A_2 is successful in winning the Game II or it successfully breaks the signature scheme, if the following conditions hold:

1. Adversary A_2 never makes the $SKQuery$ for the target sensor node identity.
2. Adversary A_2 never makes the $SQuery$ for the target identity and target message.
3. Adversary A_2 successfully forges the

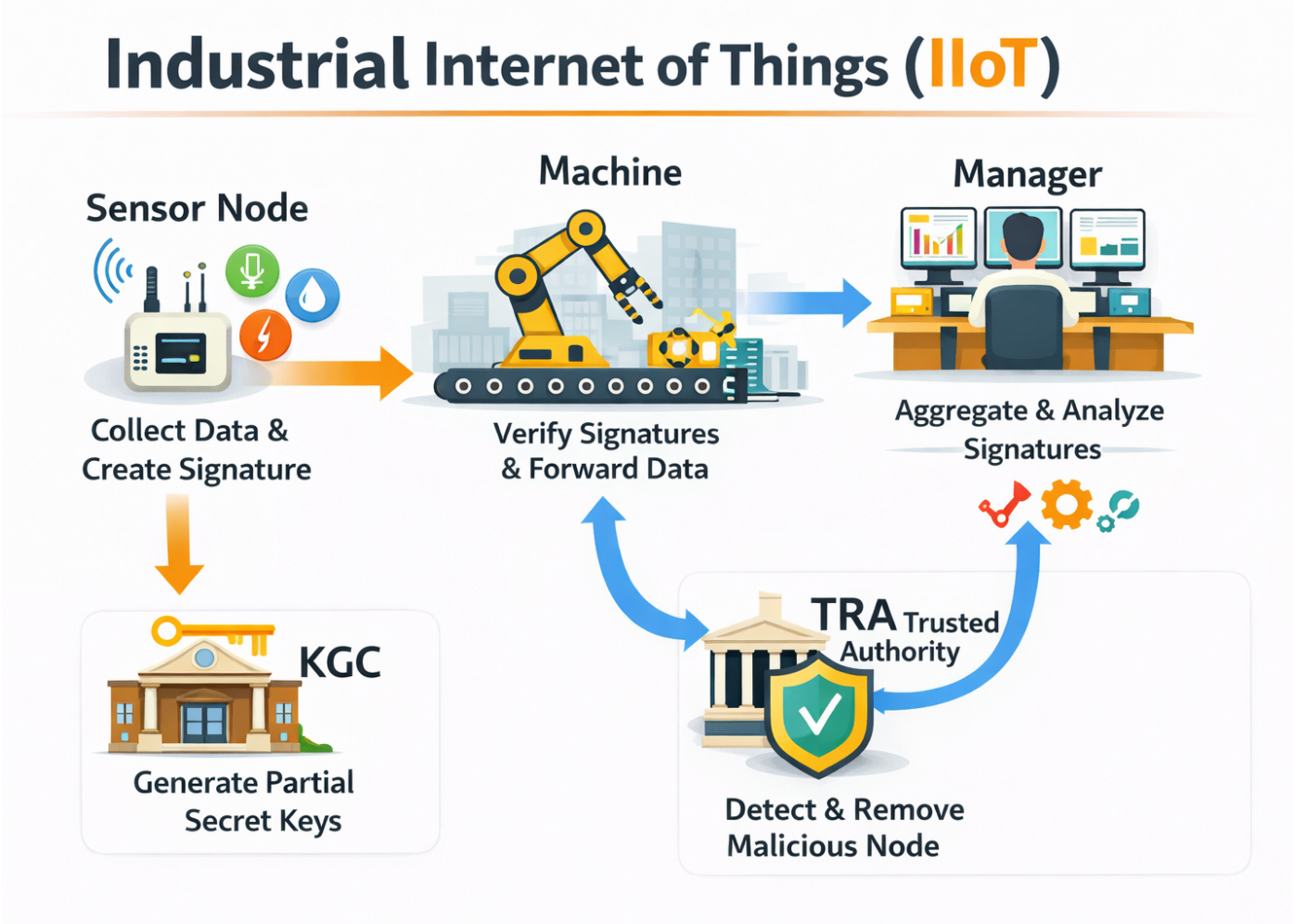

Figure 1. Network model of IIOT-CLAS architecture.

signature.

4 Cryptanalysis and Improvement of Horng et al.'s Scheme

We now prove that the scheme in [21] is vulnerable to a Type I adversary and subsequently present an improvement. Based on its capabilities, a Type I adversary can obtain a sensor node's secret key and construct a forged signature by replacing the node's public key with an arbitrary one. We then show that such a forged signature can still pass the verification process. This demonstrates that a Type I adversary can successfully forge the signature of the scheme in [21]. A brief overview of the scheme is presented in Table 3.

4.1 Type I Attack on Horng Scheme

As per the Type-I adversary capability, adversary A_1 is a malicious sensor node and has access to sensor node secret key $snsk_i$. Adversary A_1 randomly selects $y_i \in Z_p^*$ and set the new public key of sensor node SN_i by $snpk_i^* = y_i P$. Now for the $\{PseI_i, m_i\}$, adversary A_1 submits the sign request and obtains the valid signature $sig_i = (A_i, B_i)$, also with the help of Ha_{a2} finds the value h_i. Now adversary A_1 sets the forged signature $(sig_i^* = (A_i^*, B_i^*)$ such that $A_i^* = A_i - snsk_i Q$ and $B_i^* = B_i - h_i^{-1} snpk_i^*$. Verification holds as follows:

$$\begin{aligned} e(A_i^*, P) &= e(A_i - snsk_i Q, P) \\ &= e(A_i, P).e(-snsk_i Q, P) \\ &= e(X_i, P_{pkey}).e(snpk_i + h_i B_i, Q).e(-snpk_i, Q) \\ &= e(X_i, P_{pkey}).e(h_i B_i, Q) \\ &= e(X_i, P_{pkey}).e(h_i(B_i^* + h_i^{-1} snpk_i^*, Q) \\ &= e(X_i, P_{pkey}).e(h_i B_i^* + snpk_i^*, Q) \end{aligned}$$

4.2 Improved CLAS Scheme

In this section, we use certificate-less signature and aggregate signature primitives to create a novel certificateless aggregate signature scheme in which a single verification cost and aggregate

Table 3. Brief overview of Horng et al. Scheme [21].

Algorithm	Description
Setup	$params = (q, G_1, G_2, P, Q, P_{pkey}, TRA_{pkey}, Ha_{a0}, Ha_{a1}, Ha_{a2}, P_{pkey} = \alpha P, TRA_{pkey} = \beta P)$
Pseudo-identity	$PseI_{i,1} = k_i P, PseI_{i,2} = ReI_i \bigoplus Ha_{a0}(\beta PseI_{i,1}, tv_i), PseI_i = (PseI_{i,1}, PseI_{i,2}, tv_i)$
Partial secret key	$X_i = Ha_{a1}(PseI_i), snpark_i = \alpha X_i$
Sensor node key	$\gamma_i \in Z_q^*, snsk_i = \gamma_i, snpk_i = \gamma_i P$
Sign	$b_i \in Z_q^*, B_i = b_i P, h_i = Ha_{a2}(m_i, PseI_i, B_i, ts_i), A_i = snpark_i + (snsk_i + h_i b_i).Q$
Signature	$sig_i = (A_i, B_i)$
Verify	$e(A_i, P) = e(X_i, P_{pkey}).e(snpk_i + h_i B_i, Q)$
Aggregate	$A = \sum_{i=1}^{n} A_i$
Aggregate signature	$\sigma = (A, B_1, B_2, \ldots, B_n)$
Aggregate verify	$e(A, P) = e(\sum_{i=1}^{n} X_i, P_{pkey}).e(\sum_{i=1}^{n}(snpk_i + h_i B_i), Q)$

signature verification require only two bilinear pairing operations. The detailed framework of the scheme is as follows:

1. Setup

- The Trusted authority (TRA) and KGC jointly set up the system. They select a security parameter l, after which two cyclic groups G_1 and G_2 of the same order q are constructed, where q is a huge prime number, and a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ is constructed. G_1 is an elliptic curve cyclic group, with two generators P and Q, and G_2 is another multiplicative cyclic group. Three hash functions are also used in the scheme. The first hash function $Ha_{a1} : \{0,1\}^* \rightarrow G_1$ will be used for creating the partial secret key of the sensor node. The second hash function, $Ha_{a2} : \{0,1\}^* \rightarrow Z_q^*$, will be used in signature generation. The third hash function $Ha_{a0} : \{0,1\}^* \rightarrow Z_q^*$ will be used by TRA to create the pseudo-identity of the sensor node.

- KGC selects $\alpha \in Z_q^*$ and fixes it as the KGC private key and then find the value $P_{pkey} = \alpha P$ known as the KGC public key.

- TRA selects $\beta \in Z_q^*$ randomly as its private key, and this key will also be utilized to generate the sensor node's pseudo-identity. The TRA then establishes $TRA_{pkey} = \beta P$ as TRA master public key. Both these keys play a significant role in tracing a malicious sensor node.

- TRA publishes system parameters as:

$$params = (q, G_1, G_2, P, Q, P_{pkey}, TRA_{pkey}, Ha_{a0}, Ha_{a1}, Ha_{a2})$$

2. Pseudo-identity

In our scheme, we make use of the pseudo-identity of a sensor node in place of the original one. This helps in achieving the anonymity of any sensor node. The TRA generates the sensor node's pseudonymous identity and transmits it to the sensor node through a safe way. In any conflict, TRA can ascertain the legal identity of the sensor node. Thus, the privacy under conditioned sensor node is obtained as follows:

- ReI_i is the real identity of sensor node SN_i, then sensor node chooses $k_i \in Z_q^*$ randomly and calculates $PseI_{i,1} = k_i P$ and transfers $\{ReI_i, PseI_{i,1}\}$ to TRA. After receiving this TRA calculates $PseI_{i,2} = ReI_i \bigoplus Ha_{a0}(\beta PseI_{i,1}, tv_i)$. Finally TRA sends $PseI_i = \{PseI_{i,1}, PseI_{i,2}, tv_i\}$ to sensor node SN_i, where tv_i is the time of validity of pseudo-identity $PseI_i$. Our scheme achieve conditional privacy as only TRA can retrieve the real identity of the malicious sensor nodes' pseudo-identity by using its secret key as follows: $ReI_i = PseI_{i,2} \bigoplus Ha_{a0}(\beta PseI_{i,1}, tv_i)$.

3. Partial secret key

KGC generates the partial secret key for the sensor node. For this, KGC takes the pseudo-identity of the sensor node and, with the help of a hash function and its secret key, generates a partial secret key, and the key is sent to the sensor node securely as follows:

- KGC uses Ha_{a1} hash function to give $X_i = Ha_{a1}(PseI_i)$. KGC makes use of its master secret key α to produce the partial secret key $snpark_i = (\alpha X_i) \mod q$.
- Finally, KGC transfers $\{PseI_i, snpark_i\}$ to sensor node SN_i.

4. Sensor node key
In the scheme, the sensor node generates its key itself. The sensor node's signature key is derived from an amalgamation of the private key possessed by the sensor node along with a partial secret key issued by the KGC. If KGC is dishonest, this combination makes it impossible for them to forge the signature. Sensor node SN_i randomly selects $\gamma_i \in Z_q^*$ and set its private key $snsk_i = \gamma_i$ and its public key $snpk_i = \gamma_i P$.

5. Signature
This algorithm is a probabilistic algorithm. Every time a sensor node wishes to generate a signature on any identity - message pair $\{PseI_i, m_i\}$, it chooses $b_i \in Z_q^*$ to create the non deterministic signature and calculates $B_i = b_i P$. Now the sensor node calculates the hash value $h_i = Ha_{a2}(m_i, PseI_i, snpk_i, B_i, Q, t_i)$ which makes the signing key tamper resistant. (1) gives one component of the signature $sig_i = (B_i, A_i)$.

$$A_i = h_i(snpark_i + (snsk_i + b_i)P_{pkey}) \tag{1}$$

Sensor node SN_i sends the signature $sig_i = (A_i, B_i)$, as the signature on identity $PseI_i$ and message m_i along with $\{PseI_i, snpki, t_i\}$ to nearby sensor nodes and nearby verifier machine.

6. Verification
After receiving the signature sig_i, verifier machine first of all checks the freshness of time stamp. If $t_i < \delta$, where δ is the maximum time delay, then the signature is further verified else rejected immediately. Verifier calculates $X_i = Ha_{a1}(PseI_i)$ and
$h_i = Ha_{a2}(m_i, PseI_i, snpk_i, B_i, Q, t_i)$ and then calculates $W_i = X_i + snpk_i + B_i$ and verifies the signature with the help of the (2)

$$e(A_i, P) = e(h_i W_i, P_{pkey}) \tag{2}$$

7. Aggregation
If $U = \{U_1, U_2, \ldots, U_n\}$ be a set of n sensor nodes with corresponding pseudo identities $PseI = \{PseI_1, PseI_2, \ldots, PseI_n\}$, public keys $UPK = \{snpk_1, snpk_2, \ldots, snpk_n\}$ and signatures $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Manager calculates $A = \sum_{i=1}^{n} A_i$ and sends the aggregated signature $\sigma = \{A, B_1, B_2, \ldots, B_n\}$ to the verifier.

8. Aggregate verification
After receiving σ along with $\{PseI, UPK\}$, verifier calculates $h_i = Ha_{a2}(m_i, PseI_i, snpk_i, B_i, Q, t_i)$ and $X_i = Ha_{a1}(PseI_i)$ for $i = 1, 2 \ldots, n$ and then finds $W_i = X_i + snpk_i + B_i$ and validates the legitimacy of the aggregate signature utilizing (3).

$$e(A, P) = e(\sum_{i=1}^{n} h_i W_i, P_{pkey}) \tag{3}$$

Correctness

$$\begin{aligned}
e(A, P) =& e(\sum_{i=1}^{n} A_i, P) \\
=& e(\sum_{i=1}^{n} h_i(snpark_i + (snsk_i + b_i)P_{pkey}, P) \\
=& \prod_{i=1}^{n} e(h_i X_i, P_{pkey}) e((h_i snsk_i + B_i), P_{pkey}) \\
=& \prod_{i=1}^{n} e(h_i(X_i + snpk_i + B_i), P_{pkey}) \\
=& \prod_{i=1}^{n} e(h_i W_i, p_{pkey}) \\
=& e(\sum_{i=1}^{n} h_i W_i, P_{pkey})
\end{aligned}$$

5 Security analysis

We shall prove that our suggested approach is secure to withstand Type I and Type II attackers. Our scheme's security relies on how difficult the CDHP is on the elliptic curve G_1. We will show that if there is a Type I adversary A_1 with the strength to compromise our scheme in polynomial bounded time, then there is a simulator who is capable enough in solving the CDHP in polynomial bounded time. This reduction will prove Type I resistance of our scheme. In a similar manner, We will show how resilient our strategy is against a Type II threat A_2 through a reduction. We explicitly state that the security proof for our scheme is conducted within the standard model and does not rely on the forking lemma.[27]

Theorem 1 (Type I Resistant)**.** *If a Type I attacker A_1 can compromise the scheme in polynomial time, then a simulator capable of solving the CDHP in polynomial time also exists.*

Proof. Let us assume that a simulator S has an access to $\{P, aP, bP\}$. The simulator interacts with adversary A_1 to give the output abP. The adversary can make

the following query as per the requirement, and the simulator answers as per availability.

CUQuery This query basically creates sensor node key and is made whenever the public key of sensor node is required. This query, when invoked with the identity $PseI_i$, simulator S to search the list $L = \{PseI_i, snsk_i, snpk_i, snpark_i\}$ and gives back the $snpk_i$ as per availability. In case, $snpk_i$ is not available in the list L, it asks for PSKQuery, SKQuery, and CUQuery queries and insert the corresponding term in the list L.

SKQuery When adversary A_1 asks the query for $PseI_i$, simulator S searches list L and gives back the secret key as per availability.

PSKQuery This query is for partial secret key of the sensor node. When this query is asked, the simulator searches the list L and returns the corresponding partial secret key to the attacker. In case, list L does not have partial secret key corresponding to the identity $PseI_i$, and $PseI_i$ is the target identity, then $\perp$ is returned to the attacker and the game is terminated, otherwise, the simulator chooses $q_i \in Z_q^*$ creates the partial secret key $snpark_i = q_i bP$ and returns to simulator.

SQuery This is basically sign query. When this query is asked for the target identity, simulator terminates the game. In other case, when adversary makes the query with $\{m_i, PseI_i\}$, simulator chooses $b_i \in Z_q^*$ randomly and sets $B_i = b_i P$ and finds $h_i = Ha_{a2}(m_i, PseI_i, snpk_i, B_i, Q, t_i)$ creates the signature $A_i = h_i(snpark_i + (snsk_i + b_i)P_{key}$ and returns the valid signature (B_i, A_i) to the attacker.

Now A_1 and simulator S jointly solve the CDHP as follows. Simulator sets $P_{pkey} = aP$. Now the signature $sig_i = (A_i, B_i)$ that satisfies (2) is given by (1). Adversary A_1 and simulator S jointly produce the solution of CDHP as:

$$h_i^{-1} q_i^{-1}(A_i - (x_i + b_i)P_{pkey}) = abP.$$

□

Theorem 2 (Type II resistant). *If a Type II adversary A_2 has the caliber to compromise the proposed scheme within polynomial bounded time, then we can have a simulator with the strength of solving the CDHP in polynomial time.*

Proof. Let us assume that a simulator S has an access to the CDHP problem given by $\{P, aP, bP\}$. The simulator interacts with adversary A_2 to give the output abP. The adversary can make the following query as per the requirement, and the simulator answers as per availability.

CUQuery This query basically creates sensor node key and is made whenever the public key of sensor node is required. This query, when invoked with the identity $PseI_i$, simulator S searches the list $L = \{PseI_i, snsk_i, snpk_i, snpark_i\}$ and gives back the $snpk_i$ as per availability. Otherwise, simulator sets $snpk_i = bP$ and returns to the attacker.

SKQuery When adversary A_1 asks the query for $PseI_i$, simulator S searches list L and gives back the secret key as per availability.

PSKQuery This query is for partial secret key of the sensor node. When this query is asked, the simulator searches the list L and returns the corresponding partial secret key to the attacker. In case, list L does not have partial secret key corresponding to the identity $PseI_i$, and $PseI_i$ is the target identity, then $\perp$ is returned to the attacker and the game is terminated, otherwise, the simulator chooses $q_i \in Z_q^*$ and sets $X_i = q_i P$, and creates the partial secret key $snpark_i = \alpha X_i$ and returns to simulator.

SQuery This is basically siqn query. When this query is asked for the target identity, simulator terminates the game. In other case, when adversary makes the query with $\{m_i, PseI_i\}$, simulator chooses $b_i \in Z_q^*$ randomly and sets $B_i = b_i P$ and finds $h_i = Ha_{a2}(m_i, PseI_i, snpk_i, B_i, Q, t_i)$ creates the forged signature $A_i = h_i \alpha(X_i + snpk_i + B_i)$ and returns the forged signature (B_i, A_i) to the attacker.

Now A_2 and S jointly solve the CDHP as follows. Adversary makes $Sign(PseI_i, m_i)$ and S returns $sig_i = (A_i, B_i)$. Simulator sets $P_{pkey} = aP$. Also note that A_2 has access to α. Now the signature $sig_i = (A_i, B_i)$ that satisfies (2) is given by (1). Adversary A_2 and simulator S jointly produce the solution of CDHP as:

$$h_i^{-1}(A_i - \alpha(X_i + B_i)) = abP.$$

□

Table 4. Various cryptographic operations and their costs in milliseconds [31].

Operations	CostP	CostM	CostA
Cost (ms)	8.60	3.22	0.05

Table 5. Computation cost and communication cost comparison in various schemes.

Scheme	CostS (ms)	CostV (ms)	Signature size (bytes)
[28]	9.7	40.8	256
[8]	9.7	32.2	256
[29]	29.5	40.8	256
[30]	3.2	30.2	256
Ours	9.7	20.52	256

6 Results and Discussion

Now we do comparison of the performance of our scheme with that of other schemes [8, 28–30] with respect to computation costs. The costs depend on the signature creation (CostS), verification (CostV) and aggregate verification (CostVAgg) costs. To create our signature, we require three scalar multiplications, so the CostS is 3CostM $\approx$ 9.7 ms. For single signature verification, we need one scalar multiplication, two point additions and two bilinear pairing operations, and thus the CostV cost comes out to be $2\,\mathrm{CostP} + \mathrm{CostM} + 2\,\mathrm{CostA} \approx 20.52$ ms. We have listed the remaining costs numerically in Table 5. For aggregate signature verification, we require 2 bilinear pairing operations, 2n point additions and n scalar multiplications. Thus, the CostVAgg becomes 2CostP + n CostM + 2n CostA. Using this formula, we have evaluated the aggregate signature verification cost and the values have been shown in Figure 3 It is clear from the Table 5 that our computation cost is lower than other schemes. This fact is also demonstrated in Figure 2. The operations costs shown in Table 4 have been taken from [31]. the graphs clearly show that as the number of signatures increases, our scheme shows better performance than others. From the beginning, our focus was on minimizing the pairing operation, as bilinear pairing costs are costly. We have largely succeeded in reducing the costs of our scheme. In calculating the communication cost, we just find the size of each signature. We consider each group element to have a size of 128 bytes. The signature size in our scheme is twice the size of the group. The signature size of each scheme comes out to be 256 byte. This is shown in Table 5.

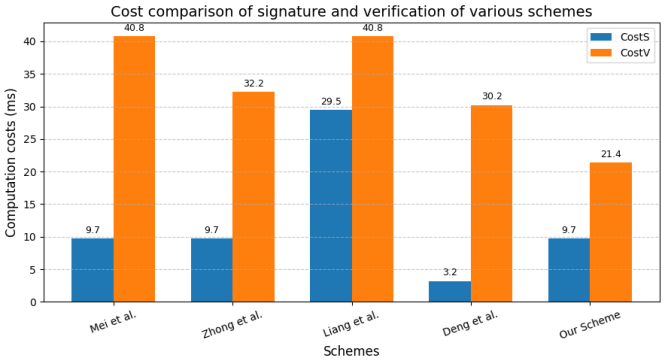

Figure 2. Signature and verification cost comparison of different schemes.

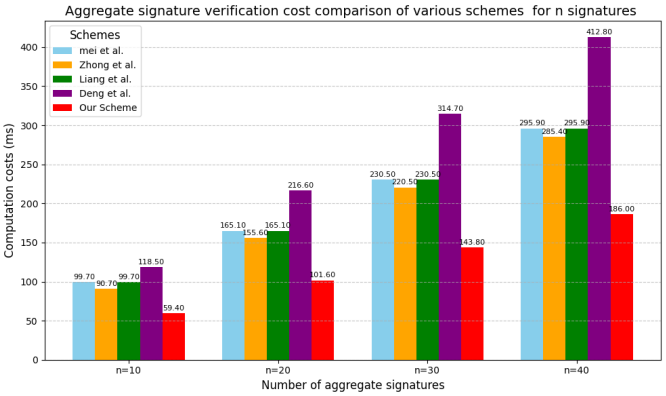

Figure 3. Aggregate signature verification cost comparison of different schemes.

7 Conclusion

This paper applies a Type I attack to the Horng et al. approach and enhances it to establish a conditional privacy-preserving framework. We also established that our scheme is powerful enough to withstand against both Type I and Type II adversaries through a brief security analysis that does not utilize the forking lemma. We then conducted a performance evaluation, demonstrating that for the validation of an individual signature, our approach only requires two pairing operations; for the verification of n signatures, it requires two pairing operations as well, and this number does not increase with increasing signers but remains fixed.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Li, S., Xu, L., & Zhao, S. (2015). The internet of things: A survey. *Information Systems Frontiers*, *17*(2), 243-259. [CrossRef]

[2] Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644-654. [CrossRef]

[3] Shamir, A. (1984, August). Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques* (pp. 47-53). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[4] Zhang, L., & Zhang, F. (2009). A new certificateless aggregate signature scheme. *Computer Communications*, *32*(6), 1079-1085. [CrossRef]

[5] Al-Riyami, S. S., & Paterson, K. G. (2003, November). Certificateless public key cryptography. In *International conference on the theory and application of cryptology and information security* (pp. 452-473). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[6] Raya, M., & Hubaux, J. P. (2007). Securing vehicular ad hoc networks. *Journal of computer security*, *15*(1), 39-68. [CrossRef]

[7] He, D., Zeadally, S., Xu, B., & Huang, X. (2015). An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, *10*(12), 2681-2691. [CrossRef]

[8] Zhong, H., Han, S., Cui, J., Zhang, J., & Xu, Y. (2019). Privacy-preserving authentication scheme with full aggregation in VANET. *Information Sciences*, *476*, 211-221. [CrossRef]

[9] Wang, H., Wang, L., Zhang, K., Li, J., & Luo, Y. (2022). A conditional privacy-preserving certificateless aggregate signature scheme in the standard model for VANETs. *IEEE Access*, *10*, 15605-15618. [CrossRef]

[10] Li, J., Yuan, H., & Zhang, Y. (2016). Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. *Cryptology ePrint Archive*.

[11] Yang, X., Wang, W., Tian, T., & Wang, C. (2023). Cryptanalysis and improvement of a blockchain-based certificateless signature for IIoT devices. *IEEE Transactions on Industrial Informatics*, *20*(2), 1884-1894. [CrossRef]

[12] Shim, K. A. (2024). A secure certificateless signature scheme for cloud-assisted Industrial IoT. *IEEE Transactions on Industrial Informatics*, *20*(4), 6834-6843. [CrossRef]

[13] Boneh, D., Gentry, C., Lynn, B., & Shacham, H. (2003, May). Aggregate and verifiably encrypted signatures from bilinear maps. In *International conference on the theory and applications of cryptographic techniques* (pp. 416-432). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[14] Peter, O., Pradhan, A., & Mbohwa, C. (2023). Industrial internet of things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in emerging economies. *Procedia Computer Science*, *217*, 856-865. [CrossRef]

[15] Hou, K., Diao, X., Shi, H., Ding, H., Zhou, H., & De Vaulx, C. (2023). Trends and challenges in AIoT/IIoT/IoT implementation. *Sensors*, *23*(11), 5074. [CrossRef]

[16] Olanrele, O., Ajagbe, S., Ilori, A., & Adeyemi, O. (2025). The industrial internet of things (IIoT): Overview, architecture, challenges, and possible solutions. *Computational Intelligence in Industry 4.0 and 5.0 Applications*, 37-60.

[17] Gentry, C., & Ramzan, Z. (2006). Identity-based aggregate signatures. *International Workshop on Public Key Cryptography*, 257-273. [CrossRef]

[18] Gong, Z., Long, Y., Hong, X., & Chen, K. (2007, July). Two certificateless aggregate signatures from bilinear maps. In *Eighth ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD 2007)* (Vol. 3, pp. 188-193). IEEE. [CrossRef]

[19] Huang, X., Mu, Y., Susilo, W., Wong, D. S., & Wu, W. (2007, July). Certificateless signature revisited. In *Australasian conference on information security and privacy* (pp. 308-322). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[20] Xiong, H., Guan, Z., Chen, Z., & Li, F. (2013). An efficient certificateless aggregate signature with constant pairing computations. *Information Sciences*, *219*, 225-235. [CrossRef]

[21] Horng, S., Tzeng, S., Huang, P., Wang, X., Li, T., & Khan, M. (2015). An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. *Information Sciences*, *317*, 48-66. [CrossRef]

[22] Thumbur, G., Rao, G., Reddy, P., Gayathri, N., Reddy, D., & Padmavathamma, M. (2020). Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks.

IEEE Internet of Things Journal, *8*(3), 1908-1920. [CrossRef]

[23] Kar, J., Liu, X., & Li, F. (2021). CL-ASS: An efficient and low-cost certificateless aggregate signature scheme for wireless sensor networks. *Journal of Information Security and Applications*, *61*, 102905. [CrossRef]

[24] Vashishtha, P., & Kumari, S. (2025). Cryptanalysis and improvement on low-cost CL-ASS for wireless sensor networks. *International Conference on Network Security and Blockchain Technology*, 19-29. [CrossRef]

[25] Xiong, W., Wang, R., Wang, Y., Wei, Y., Zhou, F., & Luo, X. (2022). Improved certificateless aggregate signature scheme against collusion attacks for VANETs. *IEEE Systems Journal*, *17*(1), 1098-1109. [CrossRef]

[26] Vashishtha, P., & Kumari, S. (2025). An improved collusion resistant certificateless aggregate signature scheme for VANETs. *International Conference on Mathematics and Computing*, 13-24. [CrossRef]

[27] Pointcheval, D., & Stern, J. (1996). Security proofs for signature schemes. *International Conference on the Theory and Applications of Cryptographic Techniques*, 387-398. [CrossRef]

[28] Mei, Q., Xiong, H., Chen, J., Yang, M., Kumari, S., & Khan, M. (2020). Efficient certificateless aggregate signature with conditional privacy preservation in IoV. *IEEE Systems Journal*, *15*(1), 245-256. [CrossRef]

[29] Liang, Y., & Liu, Y. (2022). Analysis and improvement of an efficient certificateless aggregate signature with conditional privacy preservation in VANETs. *IEEE Systems Journal*, *17*(1), 664-672. [CrossRef]

[30] Deng, L., Wen, J., Gao, Y., Wang, N., Huang, H., & Li, S. (2024). Certificateless aggregate signature scheme with security proofs in the standard model suitable for Internet of Vehicles. *IEEE Internet of Things Journal*, *11*(17), 28765-28773. [CrossRef]

[31] Chen, C., Li, Z., Das, A., Chaudhry, S., & Lorenz, P. (2024). Provably secure authentication scheme for fog computing-enabled intelligent social internet of vehicles. *IEEE Transactions on Vehicular Technology*, *73*(9), 13600-13610. [CrossRef]

Pushpendra Kumar Vashishtha Received the M.Tech. degree in Computer Science and Data Processing from I.I.T. Kharagpur, West Bengal-India. Received the M.Sc. degree in Mathematics from C.C.S. University, Meerut-India. He is currently working as an Assistant Professor in Department of Mathematics, DAV (PG) College, Muzaffarnagar and also working as a Research Scholar in Department of Mathematics, Chaudhary Charan Singh University, Meerut. (Email: pktha.iitkgp@gmail.com)

Saru Kumari is an Associate Professor with the Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh, India. She received her PhD in Mathematics in 2012 from Chaudhary Charan Singh University, Meerut, UP, India. She is the recipient of the India Research Excellence-Citation Awards-Women in Research-2023 by Clarivate Analytics. She has published more than 380 research papers in reputed international journals and conferences, including more than 330 research papers in various SCIE-indexed journals such as IEEE TDSC, IEEE TII, IEEE JBHI, IEEE T-ITS, IEEE TCSS, IEEE TCE, IEEE TGCN, IEEE IoTJ, Information Fusion, ACM TOIT, ACM TOMM, etc. She received the Best Paper award from the Journal of Network and Computer Applications, Elsevier, in 2020, the IEEE Consumer Electronics Magazine in 2022, and Vehicular Communication in 2022. She is a Senior Editor in IEEE T-ITS. She is on the editorial board of more than a dozen International Journals of high repute, under IEEE, Elsevier, Springer, Wiley, and others, including SCI and SCIE journals such as IEEE T-ITS, IEEE SJ; CSI, Elsevier; AEÜ - IJEC, Elsevier; IJCS, Wiley; CPE, Wiley; TELS, Springer; ETT, Wiley, etc. She has served as the Guest Editor of many special issues in reputed SCIE Journals under IEEE, Elsevier, Springer, and Wiley. She has been involved in the research community as a Technical Program Committee (TPC) member or PC chair for more than a dozen international conferences of high repute. She is also a reviewer of dozens of reputed Journals, including SCI-Indexed Journals, under IEEE, Elsevier, Springer, Wiley, Taylor & Francis, etc. Her research interests include Applied Cryptography, Information Security, Internet of Things, Information Fusion, Blockchain Technology, Security, and Artificial Intelligence.