Course Project M'24 - Research in Information Security

Improvised Pairing-Free Certificateless Aggregate Signature in Healthcare Wireless Medical Sensor Networks

Authors
Shreyas Adiga (CSD), 2022111010
Maneesh Manoj (CSD), 2022111011
Sumit Kumar (CSD), 2022111012

05th November 2024

1 Abstract

The healthcare wireless medical sensor network is gradually changing the traditional mode of medical treatments with the rapid development of Internet of Things. Specifically, patients' healthcare data can be continuously collected by medical sensor nodes and transmitted to the medical specialists for disease monitoring, diagnosis and treatments. Recently, due to its advantages of low computational and communication overheads in a multiuser environment, the certificateless aggregate signature (CLAS) scheme has been adopted to prevent the sensitive health care data from being tampered and damaged, thereby ensuring the integrity and authenticity of data. In order to further improve the efficiency of CLAS schemes for the sensor nodes with limited resources, several CLAS schemes without bilinear pairing have been proposed. In this article, we analyze the security of a pairing-free CLAS scheme proposed by Zhan et al. [IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5973-5984, 2021] by pointing out that their scheme is insecure against some specefic adversaries. After that, we introduce an improved scheme to solve the security vulnerability. The security proofs show that our improved scheme solves the vulnerability issues of Zhan et al scheme while keeping the same basic structure as Zhan et al scheme, which was generally secure under the ECDLP assumption in the random oracle model with some very specific exceptions as mentioned in this paper.

2 Keywords

Internet of Things (IoT), Industrial Internet of Things (IIoT), Certificateless aggregate signature (CLAS), Elliptic Curve Discrete Logarithm Problem (ECDLP), Chosen Message Attacks (CMAs), Healthcare wireless medical sensor network (HWMSN), Master Secret Key, Public key, Private key, Partial private key

3 Introduction

The Internet of Things (IoT) is defined as a network where physical objects are embedded with sensors and actuators, connected through wireless and wired networks, enabling seamless interaction and information exchange between objects in the physical and virtual world [4]. IoT has lots of applications. The deployment of Internet of Things (IoT) promotes the notion of Industrial Internet of Things (IIoT), which will be used to continuously improve production process in actual applications, and can also employed to improve efficiency and quality, to reduce cost and consumption. IIoT is a new type of infrastructure, application mode, and industrial ecology that integrates the new generation of information and communication technology. In this project, we will stick to application of IoT in Healthcare wireless medical sensor network (HWMSN).

3.1 Background

HWMSN is a significant application of IoT in the medical field. A typical HWMSN system consists of various medical sensor nodes (MSNs), a central control agency and a medical center. Several medical sensors are placed on the body surface of patients or implanted into the body to monitor their medical information and vital signs in real time, including respiration, heartbeat, temperature, blood pressure, blood glucose, blood oxygen saturation, etc. Patients' medical data is transmitted from the sensors to the central control for packaging and integration, then sent to the medical center. Healthcare professionals make diagnoses and put forward the views of the treatments for patients according to these medical data [2]. HWMSN helps improve hospital resource allocation as well as provide a smoother experience for the patients. Since the details of patient and their condition is being transmitted over a network, security of this information is of concern. The personal information of the patients should not be retrievable from a communication and the medical information being transmitted should not be tamper-able, otherwise this will lead to misdiagnosis. One way to solve this issue is to use Certificate based signature scheme, which ensures integrity of data through the means of public verification of signatures [5]. But the cost of operation for an ordinary signature scheme is too high for a large number of nodes. Recently, due to its advantages of low computational and communication overheads in a multiuser environment, the certificateless aggregate signature (CLAS) scheme has been adopted to prevent the sensitive healthcare data from being tampered and damaged, thereby ensuring the integrity and authenticity of data [2].

On a high level, a CLAS scheme is composed of seven algorithms:

MasterKeyGen: Given security parameter k, outputs master secret key msk and system parameters params.

PartialKeyGen: Given params, msk, and real identity RID_i of sensor node MSN_i, outputs partial private key D_i and pseudo identity ID_i for MSN_i.

UserKeyGen: Given ID_i of MSN_i, outputs public/secret key pair $(\mathrm{pk}_i, \mathrm{sk}_i)$ for MSN_i.

Sign: Given ID_i, secret key sk_i, partial private key D_i, and message m_i for MSN_i, outputs a signature σ_i on m_i.

Verify: Given signature σ_i, message m_i, and public key pk_i under ID_i of MSN_i, outputs *True* if σ_i is valid; otherwise, $\perp$.

Aggregate: Given n signatures $\{\sigma_i\}_{i=1}^n$ and messages $\{m_i\}_{i=1}^n$, outputs an aggregate signature σ on $\{m_i\}_{i=1}^n$.

AggregateVerify: Given aggregate signature σ, messages $\{m_i\}_{i=1}^n$, and public keys $\{\mathrm{pk}_i\}_{i=1}^n$ under $\{\mathrm{ID}_i\}_{i=1}^n$, outputs *True* if σ is valid; otherwise, $\perp$.

There is the following assumption of intractability problem..

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given a cyclic group G of points on an elliptic curve over a finite field, a generator point $P \in G$, and another point $Q \in G$, find an integer $z \in \mathbb{Z}_q^*$ such that $Q = zP$. Here, G is the set of all points on the elliptic curve that form a group under elliptic curve point addition, and q is the order of G.

3.2 System Model

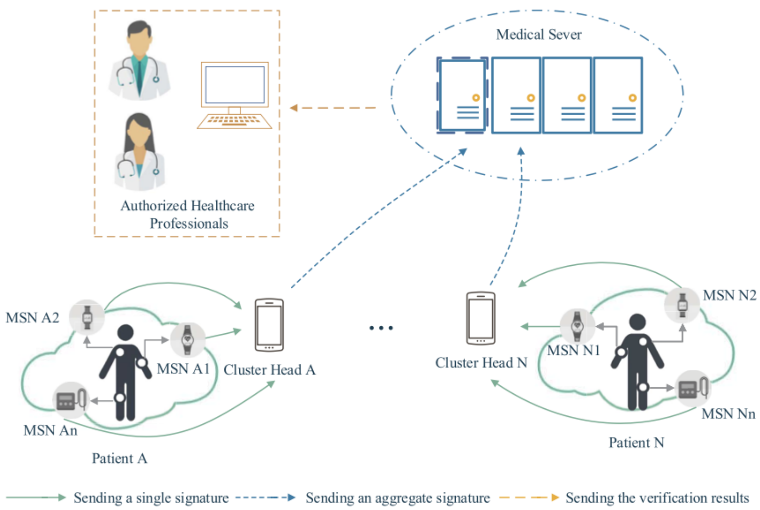

Figure 1: Framework of CLAS for HWSN [2]

There are four parties involved in a CLAS scheme for HWMSN as been in Fig 1:

1) **MSNs:** Resource-limited devices on or within a patient's body that collects healthcare data. Each MSN signs its message with a secret key before transmitting it to the cluster head (CH).

2) **CH:** Each patient's MSNs correspond to a CH responsible for data preprocessing. After receiving messages and signatures, the CH aggregates signatures and integrates messages, then sends both to the medical server (MS).

3) **MS:** Responsible for receiving and verifying the aggregated messages. MS uses the public keys of MSNs to verify the aggregate signature. If valid, the data is forwarded to authorized healthcare providers (AHPs).

4) **AHP:** Medical professionals who analyze patient data to make diagnoses and treatment plans.

3.3 Security Model

Existential Unforgeability against CMAs (Chosen Message Attacks): A CLAS scheme is secure if it resists two types of adversaries:

1) **Type I Adversary**: An external adversary capable of launching public key replacement attacks. This adversary can compromise a sensor node's secret key or replace its public key, but cannot access the master secret key or partial private keys.

2) **Type II Adversary**: An internal adversary (malicious MS) who possesses the master secret key but cannot compromise or replace sensor nodes' secret or public keys.

3.4 Research Contributions

- We review the CLAS scheme proposed by Zhan et al [2].

- We analyze and describe forgery attacks based on the methods outlined in the work of Z. Qiao et al [1] and and K A Shim et al [3].
- We propose a defense mechanism against forgery attacks within the context of the CLAS scheme.

4 Related Work

Gayathri et al. [6] constructed an efficient and secure certificateless aggregate scheme without pairing for HWMSN. Their scheme greatly improves the efficiency of signing and verification, and reduces the communication overhead of transmitting signatures while claiming to be secure. However, their solution has a fatal security hole. Liu et al. [7] put forward effective attack methods to prove that Gayathri et al.'s CLAS scheme is insecure against two kinds of attacks. Furthermore, Liu et al. gave an improved CLAS to solve the security problems. However even this scheme had some fatal issues which were highlighted by Zhan et al [2], who also provided a improvised solution which claimed to be existential unforgeable against CMAs. Qiao et al's [1] paper on certificate based aggregate signature scheme, pointed out a particular attack algorithm on the Zhan et al paper, exposing a vulnerability due to the mathematical construct of the scheme. Qiao et al left the paper at this point, without providing a replacement scheme. KA Shim [3], in her paper has pointed out few more security flaws with the scheme and provided a one stop solution for all the issues, but without any security proofs or explanation as to how her scheme is more secure.

5 Zhan et. al CLAS Scheme

5.1 Description

The scheme provided by Zhan et. al is improvement of the CLAS scheme provided by Liu et.al. It is based on the ECDLP assumption to solve the security issue of Liu et al.'s scheme [2].

1. **MasterKeyGen**: Given a security parameter k, MS selects a group G of prime order q and a generator P. Then, MS randomly selects $s \in \mathbb{Z}_q^*$ as the master secret key, and sets $P_{\text{pub}} = sP$, chooses four secure hash functions H, H_1, H_2, H_3, where $H : G \times G \to \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \times G \times G \to \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times G \times \{0,1\}^* \times G \to \mathbb{Z}_q^*$, and $H_3 : \{0,1\}^* \times \{0,1\}^* \times G \times \{0,1\}^* \to \mathbb{Z}_q^*$. Finally, the system parameters are params $= (G, q, P, P_{\text{pub}}, H, H_1, H_2, H_3)$ and the master secret key is s.
2. **PartialKeyGen**: Given the public parameters $params$, the master secret key s, and the real identity RID_i of a MSN_i, MS randomly selects $r_i \in \mathbb{Z}_q^*$ and computes $R_i = r_i P$, and $\text{ID}_i = \text{RID}_i \oplus H(r_i P_{\text{pub}}, T_i)$, $h_{1i} = H_1(\text{ID}_i, R_i, P_{\text{pub}})$ and $d_i = r_i + sh_{1i} \mod q$, where T_i denotes the pseudo identity ID_i validity time period. Then, MS sets the **partial private key** as $D_i = (d_i, R_i)$ and sends (ID_i, T_i, D_i) to the MSN_i secretly. The MSN_i verifies the validity of the partial private key by checking whether $d_i P = R_i + h_{1i} P_{\text{pub}}$ holds.

3. **UserKeyGen**: The MSN_i with ID_i randomly selects $x_i \in \mathbb{Z}_q^*$. Then, the secret key of the MSN_i is set as $\mathrm{sk}_i = (x_i, d_i)$, and the corresponding public key is set as $\mathrm{pk}_i = (X_i, R_i) = (x_i P, r_i P)$.

4. **Sign**: The MSN_i signs a message m_i at time t_i as follows.

   a) Choose a random value $y_i \in \mathbb{Z}_q^*$ and compute $Y_i = y_i P$.

   b) Compute $u_i = H_2(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i, Y_i)$ and $h_{3i} = H_3(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i)$.

   c) Compute $w_i = [u_i y_i + h_{3i}(x_i + d_i)] \mod q$.

   d) Output $\sigma_i = (Y_i, w_i)$ as the signature on $m_i||t_i$.

5. **Verify**: The CH verifies a signature σ_i on $m_i||t_i$ with the public key pk_i on ID_i as follows.

   a) Compute $h_{1i} = H_1(\mathrm{ID}_i, R_i, P_{\mathrm{pub}})$, $u_i = H_2(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i, Y_i)$, and $h_{3i} = H_3(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i)$.

   b) Accept the signature if

   $$w_i P - u_i Y_i = h_{3i}(X_i + R_i + h_{1i} P_{\mathrm{pub}})$$

   holds.

6. **Aggregate**: Given n signature $\{\sigma_i : i = 1, \ldots, n\}$ on n messages $\{m_i||t_i, i = 1, \ldots, n\}$ form n MSNs, the CH generates an aggregate signature as follows.

   a) Compute $u_i = H_2(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i, Y_i)$, $i = 1, \ldots, n$.

   b) Compute $U = \sum_{i=1}^{n} u_i Y_i$.

   c) Compute $w = \sum_{i=1}^{n} w_i$.

   d) Output the aggregate signature $\sigma = (U, w)$.

7. **AggregateVerify**: Given an aggregate signature σ on $\{m_i||t_i, i = 1, \ldots, n\}$, and n public keys $(\mathrm{pk}_i : i = 1, \ldots, n)$ on identities $\{\mathrm{ID}_i : i = 1, \ldots, n\}$, MS performs the following operations.

   a) Compute $h_{1i} = H_1(\mathrm{ID}_i, R_i, P_{\mathrm{pub}})$, and $h_{3i} = H_3(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i)$, for $i = 1, \ldots, n$.

   b) Accept the aggregate signature if

   $$wP - U = \sum_{i=1}^{n} h_{3i}(X_i + R_i + h_{1i} P_{\mathrm{pub}})$$

   holds.

   c) Correctness:

   $$wP - U = \sum_{i=1}^{n} w_i P - \sum_{i=1}^{n} u_i Y_i,$$

   $$= \sum_{i=1}^{n} (u_i Y_i + h_{3i}(X_i + d_i)P) - \sum_{i=1}^{n} u_i Y_i,$$

   $$= \sum_{i=1}^{n} h_{3i}(X_i + d_i)P,$$

$$= \sum_{i=1}^{n} h_{3i}(X_i + R_i + h_{1i}P_{pub}).$$

5.2 Cryptanalysis / Attack Model

The CMA security model for CLAS schemes consists of four games, but we will be looking at only one game which is relevant to type 1 attack. Before delving into the details of game, we introduce the following oracles provided by the challenger that adversaries can query. [2]

1. **Create User Oracle** $\mathcal{O}_{CU}(\mathrm{ID}_i)$: When adversaries query this oracle, the challenger runs *UserKeyGen*$(\mathrm{ID}_i) \rightarrow (pk_i, sk_i)$ and *PartialKeyGen*$(\mathrm{msk}, \mathrm{ID}_i) \rightarrow D_i$. Then, the challenger records $(pk_i, sk_i, D_i, \mathrm{ID}_i)$ in a list $\mathcal{L}$ and returns the public key pk_i.

2. **Secret Key Oracle** $\mathcal{O}_{SK}(\mathrm{ID}_i)$: When adversaries query this oracle, the challenger finds the tuple $(pk_i, sk_i, D_i, \mathrm{ID}_i)$ from the list $\mathcal{L}$, then returns the secret sk_i as the query result.

3. **Partial Private Key Oracle** $\mathcal{O}_{PPK}(\mathrm{ID}_i)$: When adversaries query this oracle, the challenger searches the list $\mathcal{L}$ to find $(pk_i, sk_i, D_i, \mathrm{ID}_i)$. Then, the challenger returns the partial private key D_i as the query result.

4. **Replace Key Oracle** $\mathcal{O}_{RK}(\mathrm{ID}_i, pk_i', sk_i')$: When adversaries query this oracle, the challenger finds $(pk_i, sk_i, D_i, \mathrm{ID}_i)$ from the list $\mathcal{L}$ and replaces this record with $(pk_i', sk_i', D_i, \mathrm{ID}_i)$.

5. **Sign Oracle** $\mathcal{O}_S(m_i, \mathrm{ID}_i)$: When adversaries query this oracle, the challenger executes as follows.

   (a) If there is no record about ID_i in the list $\mathcal{L}$, return a symbol $\perp$ as the result.

   (b) Otherwise, find the current public/secret key pair from the list $\mathcal{L}$, and return the result of running $\mathrm{Sign}(ID_i, sk_i, D_i, m_i)$.

Game: In this game, A_1 is a probability polynomial time (PPT) Type I adversary.

Setup: In this phase, the challenger C_1 executes *MasterKeyGen* with a security parameter k to produce the master secret key *msk* and system parameters *params*. Then, C_1 keeps *msk* secretly and sends *params* to A_1.

Query: In the query phase, the adversary A_1 makes queries in the oracles $\mathcal{O}_{CU}$, $\mathcal{O}_{SK}$, $\mathcal{O}_{PPK}$, $\mathcal{O}_{RK}$, and $\mathcal{O}_S$.

Forgery: In the final phase, A_1 chooses a target sensor node MSN_i^* with the identity ID_i^* and the public key pk_i^*, then outputs σ^* as a forged signature on m_i^*. A_1 wins the game if the result of $\mathrm{Verify}(\sigma^*, m_i^*, pk_i^*, \mathrm{ID}_i^*)$ is *True* and

1. $\mathcal{O}_S(m_i^*, \mathrm{ID}_i^*)$ has never been queried;

2. $\mathcal{O}_{PPK}(\mathrm{ID}_i^*)$ has never been queried.

As per **Z. Qiao et al's paper** [1], the Zhan et. al scheme is vulnerable against type 1 attack following the below algorithm:

Algorithm 1

1. The public key $pk_i = (X_i, R_i)$ of MSN with identity id_i is replaced with $pk'_i = (X'_i, R_i)$ by A_1, where $X'_i = -(R_i + h_{1i}P_{Pub})$ and $h_{1i} = H_1(id_i, R_i, P_{Pub})$.

2. A valid forged signature $\sigma'_i = (Y'_i, w'_i)$ on $m_i \parallel t'_i$ can be created by A_1 through the following operations.

   (a) Choose $y'_i \in \mathbb{Z}_q^*$ and compute $Y'_i = y'_i P$.

   (b) Compute $u'_i = H_2(m_i, id_i, pk'_i, t'_i, Y'_i)$ and $w'_i = u'_i y'_i$.

   (c) Output $\sigma'_i = (Y'_i, w'_i)$.

3. $\sigma'_i = (Y'_i, w'_i)$ is a valid signature, because this equation

$$w'_i P - u'_i Y'_i = h_{3i}(X'_i + R_i + h_{1i}P_{Pub}) = 0$$

   holds, where $u'_i = H_2(m_i, id_i, pk'_i, t'_i, Y'_i)$, $h_{1i} = H_1(id_i, R_i, P_{Pub})$ and $h'_{3i} = H_3(m_i, id_i, pk'_i, t'_i)$.

As per **K. A Shim's paper** [3], the Zhan et. al scheme is vulnerable against type 1 attack following the below algorithm:
Algorithm 2

1. Generate a New Public Key: A_1 picks $\alpha, \beta \in \mathbb{Z}_q^*$ and calculates

$$R'_j = \beta P, \quad h'_{1j} = H_1(ID_j, R'_j, P_{\text{pub}}),$$

$$X'_j = \alpha P - h'_{1j} P_{\text{pub}}.$$

   Then, A_1 sets a new public key $pk'_j = (X'_j, R'_j)$.

2. Replace the Public Key: A_1 replaces the public key pk_j of ID_j with the new public key pk'_j.

3. **Forgery**: After that, A_1 can produce a valid signature of a message m_j under the target identity ID_j, with the new public key $pk'_j = (X'_j, R'_j)$ as follows:

   (a) Select $y'_j \in \mathbb{Z}_q^*$ and compute $Y'_j = y'_j P$ and

$$h'_{3j} = H'_3(m_j, ID_j, pk'_j, t_j),$$

$$u'_j = H_2(m_j, ID_j, pk'_j, t'_j, Y'_j),$$

$$w'_j = u'_j y'_j + h'_{3j}(\alpha + \beta) (\mod q).$$

   (b) Output $\sigma' = (Y'_j, w'_j)$ as a signature forgery.

4. **Validity**: The signature $\sigma' = (Y'_j, w'_j)$ of m_j for ID_j with the public key pk'_j is valid: it passes the verification equation below

$$w'_j P - u'_j Y'_j = h'_{3j}(X'_j + R'_j + h'_{1j} P_{pub}).$$

   since

$$w'_j P - u'_j Y'_j = [u'_j y'_j + h'_{3j}(\alpha + \beta)]P - u'_j y'_j P$$

$$= h'_{3j}(\alpha P + \beta P),$$

$$h'_{3j}(X'_j + R'_j + h'_{1j}P_{pub})$$

$$= h'_{3j}(\alpha P - h'_{1j}P_{pub} + \beta P + h'_{1j}P_{pub})$$

$$= h'_{3j}(\alpha P + \beta P),$$

Hence, Zhan et. al's scheme fails for these algorithms and we are fixing this scheme for these particular attack algorithms.

6 Proposal

On carefully looking at the attack pattern, we notice that attackers can manipulate a public key of the target user so that the public information related to the master secret key can be removed from the verification equation. Such a removal of the public information using appropriate algebraic relations in the group makes it possible to generate valid signatures without requiring the partial private key of the target user. The cause of our type I attack on the Zhan et al scheme is due to the fact that the adversary can remove the part $h_{1j}P_{Pub}$ related to the master secret s from the verification equation by replacing the public key pk_j with a new public key pk'_j and using some algebraic relations. In general, to protect the type I attacks, CLAS schemes must be built so that the user public key cannot be produced to remove the values related to the master public key P_{pub} from the verification equation. If the attacker can produce an appropriate user public key using some algebraic relations in the group to remove the master public key from the verification equation then a forgery is always possible without using the partial private key of the target user.

Here is an updated scheme to protect the CLAS against this type of **type I attacks**:

- Note: Text in bold represents modifications made to the original scheme.

1. **MasterKeyGen**: Mostly remains the same as Zhan et al scheme, but we add a new hash function $H_4 : \{0,1\}^* \times G \times G \to \mathbb{Z}_q^*$
2. **PartialKeyGen**: Here, we change the partial key D_i, to include the hash value h_{1i}. Given the public parameters *params*, the master secret key s, and the real identity RID_i of a MSN_i, MS randomly selects $r_i \in \mathbb{Z}_q^*$ and computes $R_i = r_iP$, and $\mathrm{ID}_i = \mathrm{RID}_i \oplus H(r_iP_{\mathrm{pub}}, T_i)$, $h_{1i} = H_1(\mathrm{ID}_i, R_i, P_{\mathrm{pub}})$ and $d_i = r_i + sh_{1i} \mod q$, where T_i denotes the time interval associated to the pseudo identity ID_i. Then, MS sets the partial private key as $D_i = (d_i, R_i, h_{1i})$ and sends $(\mathrm{ID}_i, T_i, D_i)$ to the MSN_i securely. The MSN_i verifies the validity of the partial private key by checking whether $d_iP = R_i + h_{1i}P_{\mathrm{pub}}$ holds.
3. **UserKeyGen**: It remains the same as Zhan et al scheme.
4. **Sign**: Here, we modify the signature to include the binding hash value and the new hash function. The MSN_i signs a message m_i at time t_i as follows.

    a) Choose a random value $y_i \in \mathbb{Z}_q^*$ and compute $Y_i = y_iP$.

b) Compute $u_i = H_2(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i, Y_i)$, $h_{3i} = H_3(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i)$ and $\boldsymbol{h_{4i} = H_4(ID_i, R_i, P_{pub})}$

c) **Compute** $\boldsymbol{w_i = [u_i y_i + h_{3i}(x_i + d_i) + h_{4i} d_i] \mod q}$

d) Output $\sigma_i = (Y_i, w_i, \boldsymbol{h_{1i}})$ as the signature on $m_i||t_i$.

5. **Verify**: This is where the major change happens. The CH verifies a signature σ_i on $m_i||t_i$ with the public key pk_i on ID_i as follows.

   (a) Compute: $h_{1i} = H_1(ID_i, R_i, P_{pub})$

   (b) **If** $\boldsymbol{h_{1i}}$ **does not match the computed value from the public key and identity, the signature is rejected.**

   (c) Else Compute: $u_i = H_2(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i, Y_i)$, $h_{3i} = H_3(m_i, \mathrm{ID}_i, \mathrm{pk}_i, t_i)$ and $\boldsymbol{h_{4i} = H_4(\mathrm{ID}_i, R_i, P_{pub})}$.

   (d) Accept the signature if

   $$\boldsymbol{w_i P - u_i Y_i = h_{3i}(X_i + R_i + h_{1i} P_{pub}) + h_{4i}(R_i + h_{1i} P_{pub})}$$

   holds.

   (e) Correctness: Multiplying P to (4c) above , we get this.

6. **Aggregate** : It remains the same as Zhan et al scheme.

7. **Aggregate Verify**: It remains the same as Zhan et al scheme.

7 Analysis

The scheme remains very much similar to Zhan et al scheme and hence at the very least matches the security level of Zhan et al. which said that their CLAS scheme is CMA secure under the ECDLP assumption in the random oracle model. While the security was true for most cases as demonstrated in the Zhan et al paper, it had some flaws which arose due to the ability of attackers to produce an appropriate user public key using some algebraic relations in the group to remove the master public key from the verification equation. We have introduced a new hash function H_4 and have used the h_{1i} as a means of thwarting attack attempts using the previously mentioned algorithms. The addition of the new hash function thwarts the algorithm 1 type attack, where R_i is kept the same and X_i is manipulated. We will repeat the attack and show how the attack fails.

Algorithm 1

1. The public key $pk_i = (X_i, R_i)$ of MSN with identity id_i is replaced with $pk'_i = (X'_i, R_i)$ by A_1, where $X'_i = -(R_i + h_{1i} P_{Pub})$ and $h_{1i} = H_1(id_i, R_i, P_{Pub})$.

2. A forged signature $\sigma'_i = (Y'_i, w'_i, h_{1i})$ on $m_i \parallel t'_i$ if created by A_1 through the following operations.

   (a) Choose $y'_i \in \mathbb{Z}_q^*$ and compute $Y'_i = y'_i P$.

   (b) Compute $u'_i = H_2(m_i, id_i, pk'_i, t'_i, Y'_i)$ and $w'_i = u'_i y'_i$.

   (c) If the verification equation

   $$w_i P - u_i Y_i = h_{3i}(X_i + R_i + h_{1i} P_{pub}) + h_{4i}(R_i + h_{1i} P_{pub})$$

must be bypassed , the adversory must need the value of $w_i = u_i y_i + h_{3i}(x_i + d_i) + h_{4i}d_i$. Bu this can't be obtained by adversory as the term $h_{4i}d_i$, can't be obtained in any way as the adversory does not know the master secret key s and partial private key d_i. As seen in the attack model , if $w'_i = u'_i y'_i$

   (d) Output $\sigma'_i = (Y'_i, w'_i, h_{1i})$.

3. $\sigma'_i = (Y'_i, w'_i)$ is a not a valid signature, because this equation

$$w'_i P = u'_i Y'_i + h_{3i}(X'_i + R_i + h_{1i}P_{pub}) + h_{4i}(R_i + h_{1i}P_{pub})$$

   for the given X'_i comes down to

$$w'_i P = u'_i Y'_i + h_{4i}(R_i + h_{1i}P_{pub}) \neq u'_i Y'_i$$

   Hence, this protects against algorithm 1 attack.

Algorithm 2

1. Generate a New Public Key. $\mathcal{A}$ picks $\alpha, \beta \in \mathbb{Z}_q^*$ and calculates

$$R'_i = \beta P, \quad h'_{1i} = H_1(ID_i, R'_i, P_{\text{pub}}),$$

$$X'_i = \alpha P - h'_{1i}P_{\text{pub}}.$$

   Then, $\mathcal{A}$ sets a new public key $pk'_i = (X'_i, R'_i)$.

2. Replace the Public Key. $\mathcal{A}$ replaces the public key pk_i of ID_i with the new public key pk'_i.

Since we are replacing R_i here, the value of h_{1i} changes and this does not remain the same as the one generated during the PartialKey Generation and hence, the signature would be rejected.

Our proposal helps improve the security measures of Zhan et al scheme for these particular type 1 attacks, all the while maintaining the security under the ECDLP assumption in the random oracle model.

Complexity Analysis

Scheme	**Single Sign Cost**	**Single Verify Cost**	**Aggregate and Aggregate Verify Cost**
Zhan et al.'s [**2**]	$2T_{mz} + 3T_{ecsm} =$ 0.499219 ms	$3T_{mz} + 4T_{ecsm} + 3T_{ecpa} =$ 0.670432 ms	$3nT_{mz} + (3n+1)T_{ecsm} + (4n-1)T_{ecpa}$
Our Scheme	$3T_{mz} + 4T_{ecsm} =$ 0.666220 ms	$4T_{mz} + 5T_{ecsm} + 3T_{ecpa} =$ 0.837433 ms	$3nT_{mz} + (3n+1)T_{ecsm} + (4n-1)T_{ecpa}$

Table 1: Performance comparison of different schemes

Notation	Description	Running Time (in ms)
T_{mz}	Map to $\mathbb{Z}^*{}_q$ hash	0.001784
T_{ecsm}	ECC Scalar Multiplication	0.165217
T_{ecpa}	ECC Point Addition	0.001404

Table 2: Execution Time of single operation(from the Table 2 of [**2**])

8 Future Work

Our work tackles only some very specific vulnerabilities arising due to the way the model is set up. There might be more specific algorithms for which the scheme may fail. Such algorithms could be further explored and tackled. Also the Zhan et al scheme is vulnerable to universal forgery attack as shown by Kyung-Ah Shim. Our scheme can be further improved to handle this vulnerability.

9 References

1) Z. Qiao et al., "An Efficient Certificate-Based Aggregate Signature Scheme With Provable Security for Industrial Internet of Things," in IEEE Systems Journal, vol. 17, no. 1, pp. 72-82, March 2023, doi: 10.1109/JSYST.2022.3188012

2) Y. Zhan, B. Wang and R. Lu, "Cryptanalysis and Improvement of a Pairing-Free Certificateless Aggregate Signature in Healthcare Wireless Medical Sensor Networks," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5973-5984, 1 April1, 2021, doi: 10.1109/JIOT.2020.3033337.

3) K. -A. Shim, "Cryptanalysis of Compact Certificateless Aggregate Signature Schemes for HWMSNs and VANETs," in IEEE Access, vol. 12, pp. 137634-137641, 2024, doi: 10.1109/ACCESS.2024.3416954

4) Gregory Kipper,Augmented Reality, 2013

5) W. Mao, Modern Cryptography: Theory and Practice. Upper Saddle River, NJ. USA: Prentice Hall PTR, 2003.

6) N. B. Gayathri, G. Thumbur, P. R. Kumar, M. Z. U. Rahman, P. V. Reddy, and A. Layekuakille, “Efficient and secure pairing-free cer tificateless aggregate signature scheme for healthcare wireless medical sensor networks,” IEEE Internet Things J., vol. 6, no. 5, pp. 9064–9075, Oct. 2019.

7) J. Liu, L. Wang, and Y. Yu, “Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sen sor networks,” IEEE Internet Things J., vol. 7, no. 6, pp. 5256–5266, Jun. 2020.