IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2020, accepted August 24, 2020, date of publication September 10, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023093

Cryptanalysis of Two Signature Schemes for IoT-Based Mobile Payments and Healthcare Wireless Medical Sensor Networks

KYUNG-AH SHIM, (Member, IEEE)
National Institute for Mathematical Sciences, Daejeon 34047, South Korea
e-mail: kashim@nims.re.kr

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT), Korea Government under Grant NRF2020R1A2B5B02097335.

ABSTRACT Certificateless cryptography does not require any certificate for the public key authentication and users' public keys are transmitted with ciphertext/signatures or by making them available in the IoT-based public directory in a proper way. Due to these features, certificateless cryptosystems are considered as fundamental cryptographic buildingblocks to provide authenticity, integrity and non-repudiation suitable for IoT applications. Yeh proposed a transaction scheme based on a certificateless signature scheme for IoT-based mobile payments implementing on Android Pay. He showed that the CLS scheme was unforgeable against Type I and Type II adversaries under the intractability of the mathematical problem. Despite the security proofs, we show that Yeh's scheme is still insecure against both Type I and Type II adversaries. Recently, Gayathri *et al.* constructed a compact certificateless aggregate signature scheme for Healthcare Wireless Medical Sensor Networks. Their aggregate signatures are constant-size independent of the number of signers. In this paper, we show that anyone can forge certificateless aggregate signatures of their scheme on any sets of messages and identities from only publicly known information, i.e. their scheme is entirely broken. We then discuss some improvements.

INDEX TERMS Aggregate signature, certificateless signature, Type I adversary, Type II adversary, universal forgery attack.

I. INTRODUCTION

Public-key cryptography requires the public key authentication by a trusted third party called 'Certificate Authority'. The Certificate Authority plays an important role in issuing, distribution and revocation of public-key certificates corresponding to users' public keys. So, public-key cryptography has the certificate management problem such as distribution, revocation and verification overhead of the certificates. Shamir [11] introduced a notion of identity (ID)-based cryptography, where a user's public key can be computed from the user's public available identity information such as e-mail address. In ID-based cryptography, Key Generation Center (KGC) using a master secret key generates users' private which causes the key-escrow problem. Al-Riyami and Paterson [1] introduced a new notion of certificateless cryptography to solve these problems. By generating users' secret keys as the combination of secret information of KGC with a master secret key and user-chosen secret values, certificateless cryptography resolves the problems. Due to these properties, certificateless cryptosystems are considered as fundamental cryptographic buildingblocks to provide authenticity, integrity and non-repudiation for IoT applications. They do not require any certificate for the public key authentication and users' public keys are transmitted with ciphertext/signatures or by making them available in the IoT-based public directory in a proper way. Thus, the certificateless public-key cryptosystems are suitable for authenticity in IoT environments and mobile payments satisfying efficiency requirements of IoT intelligent objects.

In recent years, the payment industry such as internet service providers, mobile device manufacturers and telecommunication companies, has participated to support payments with improved protections against various types of frauds such as counterfeiting and account misuse [3]–[5], [7], [12]. Transaction security for mobile payments in IoT-based network is one of the most important issues. For secure online transactions, the payment system should be

The associate editor coordinating the review of this manuscript and approving it for publication was Aneel Rahim.

designed to achieve specific security features such as data integrity, entity authentication and non-repudiation. Recently, Yeh [13] proposed a transaction scheme based on a certificateless signature (CLS) scheme for IoT-based mobile payments implementing on Android Pay. He proved that the CLS scheme was unforgeable against Type I and Type II adversaries under the intractability of the mathematical problem. Despite the security proofs, we show that Yeh's scheme is still vulnerable to both Type I and Type II adversaries.

Healthcare Wireless Medical Sensor Networks (HWMSNs) are wireless communication networks, where medical sensor nodes embedded in a patient's body monitor, collect medical data and send the data to healthcare professionals [6], [8]. The MSN implanted on the patient's body can send patient's medical data to remote Medical Server (MS) for further processing via Cluster Head. The medical data are transmitted to the healthcare professionals by remote MS through internet, and the healthcare professionals generate the patient's medical reports. It is critical to achieve the privacy of sensitive medical data in HWMSNs [2], [10] since patient's personal health data are shared online. Millions of participants upload numerous health data to cloud severs that cause storage capacity burdens and data transmission obstacles. An aggregate signature (AS) scheme can solve these problems. The AS scheme allows to aggregate n distinct signatures on n distinct messages for n distinct users into a single signature. It can reduce computation cost for verification on numerous signatures and communication/storage overhead. Thus, certificateless aggregate signature (CLAS) schemes are suitable for secure and efficient communications in HWMSNs. There have been proposed a number of CLAS schemes for various IoT applications. However, the size of their aggregate signatures grows linearly with the number of signers in an aggregating set. Recently, Gayathri *et al.* [9] constructed a compact CLAS scheme for HWMSNs whose signature size is independent of the number of signers. The scheme was proven secure against Type I and Type II adversaries under the hardness of the mathematical problem. In this paper, we show that anyone can forge certificateless aggregate signatures of the scheme on any sets of messages and identities using only publicly known information, i.e. their scheme is entirely broken.

The rest of the paper is organized as follows. We review Yeh's CLS scheme [13] and Gayathri *et al.*'s CLAS scheme [9] in Section II. In Section III, we present Type I and Type II attacks on Yeh's CLS scheme. We then show that Gayathri *et al.*'s CLAS scheme is insecure against universal forgery attacks. We discuss some improvements in Section IV. In Section V, concluding remarks are given.

II. DESCRIPTION OF TWO SIGNATURE SCHEMES

Here, we review Yeh's CLS scheme for IoT-Based Mobile Payments [13] and Gayathri *et al.*'s CLAS scheme for HWMSNs [9].

A. REVIEW OF YEH's CLS SCHEME

Yeh's scheme [13], a pairing-free CLS scheme, is specified by the following seven algorithms:

- **Setup:** For a security parameter k, KGC does the following:
  - Select a group $\mathbb{G}$ of elliptic curve points with a prime order n and a random generator P of $\mathbb{G}$.
  - Select a master secret key $s \in \mathbb{Z}_n^*$ and calculate a master public key $PK_{KGC} = s \cdot P$.
  - Select a secure hash function
  $$H : \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{Z}_q.$$
  - At last, keep s securely and publish
  $$params =< \mathbb{G}, P, PK_{KGC}, H > .$$
- **PartialPrivateKeyExtract:** For *params*, a master secret key s and an identity ID_i of a user i, KGC generates a partial private key of the user as follows:
  - KGC chooses a random value $r_i \in_R \mathbb{Z}_n^*$ and computes
  $$R_i = r_i \cdot P, \quad h_i = H(ID_i, R_i, PK_{KGC}),$$
  $$s_i = r_i \cdot ID_i + h_i \cdot s \mod n.$$
  - Then, KGC sends a partial private key $D_i = (s_i, R_i)$ to the user i in a secure way.
  - The user can check the validity of partial private key by verifying the following equality
  $$s_i \cdot P = ID_i \cdot R_i + h_i \cdot PK_{KGC} \mod n.$$
- **SetSecretValue:** A user i chooses a random value $x_i \in_R \mathbb{Z}_n$ as a secret value.
- **SetPublicKey:** For params and x_i, a user i calculates
  $$PK_i = x_i \cdot P$$
  as a public key.
- **Sign:** For *params*, a partial private key D_i, a secret value x_i, and a message m, a user i picks a random number $t_i \in_R \mathbb{Z}_n$ and calculates
  $$T_i = t_i \cdot P, \; k_i = H(m, T_i, PK_i, h_i),$$
  $$\tau_i = t_i + k_i \cdot (x_i + s_i) \mod n.$$
  Then a signature on the message m is $\sigma_i = (R_i, T_i, \tau_i)$.
- **Verify:** Given *params*, an identity ID_i, a user public key PK_i, a message m and a signature $\sigma_i = (R_i, T_i, \tau_i)$ on m, a verifier calculates
  $$h_i = H(ID_i, R_i, PK_{KGC}),$$
  $$k_i = H(m, T_i, PK_i, h_i)$$
  and checks following equality
  $$\tau_i \cdot P = T_i + k_i \cdot (PK_i + ID_i \cdot R_i + h_i \cdot PK_{KGC}).$$
  If it holds then accept the signature.

B. REVIEW OF GAYATHRI et al.'s CLAS SCHEME

Gayathri *et al.* [9] constructed a compact CLAS scheme whose signature size is independent of the number of signers. In other words, its aggregate signature consists of two elements in G and an element in Z_q^*. Gayathri *et al.*'s scheme is composed of four entities: Cluster Head (Data Aggregator), Medical Sensor Nodes (MSNs), Medical Server (Data Center), Authorized Healthcare Professionals and seven algorithms [9]. The scheme runs as follows:

- **System Initialization:** For a security parameter $k \in Z^+$, Medical Server (MS) computes system parameters as follows:
  - Choose a group G of a prime order q, a generator P of G, selects $s \in Z_q^*$ as a master secret key and computes $P_{pub} = sP$ as a master public key.
  - Choose secure four hash functions

$$\begin{aligned} H &: G \times G \to Z_q^*, \\ H_1 &: \{0,1\}^* \times G \times G \to Z_q^*, \\ H_2 &: \{0,1\}^* \times \{0,1\}^* \times G \to Z_q^*, \\ H_3 &: \{0,1\}^* \times \{0,1\}^* \times G \times \{0,1\}^* \to Z_q^*, \\ H_4 &: \{0,1\}^* \times \{0,1\}^* \times G \times \{0,1\}^* \to Z_q^*. \end{aligned}$$

  - Publish the system parameters as

$$params = (q, G, P, P_{pub}, H, H_1, H_2, H_3, H_4)$$

  and keep the master secret key secure.
- **Partial Private Key Gen:** From a master secret s and *params*, MS generates a sensor node's partial private key as:
  - For a real identity RID_i of a sensor node, MS selects a random $r_i \in Z_q^*$ and calculates $R_i = r_iP$, generates a pseudo identity

$$ID_i = RID_i \oplus H(r_iP_{pub}, T_i),$$

  where T_i is a time period corresponding to the pseudo identity.
  - MS computes $h_{1i} = H_1(ID_i, R_i, P_{pub})$ and

$$d_i = r_i + sh_{1i} \ \ mod\ q.$$

  Then MS sets $D_i = (d_i, R_i)$ as a partial private key of the sensor node and transmits (ID_i, T_i, D_i) to the sensor node in a secure way.
  - If $d_iP = R_i + h_{1i}P_{pub}$ holds, then the sensor node accepts the partial private key D_i for ID_i at T_i.
- **Sensor Node's Public/Secret Key Pair Gen:** From D_i and ID_i, a sensor node generates a public key and a secret key as:
  - Select a random $x_i \in Z_q^*$ and compute $X_i = x_iP$.
  - Set $PK_i = (X_i, R_i)$ and $SK_i = (d_i, x_i)$ as a secret key and a public key, respectively.
- **Signature Generation:** Using $SK_i = (d_i, x_i)$, ID_i and *params*, a sensor node computes the signature as:
  - Select $y_{1i}, y_{2i} \in Z_q^*$, a current time stamp t_i, and calculate Y_{1i}, Y_{2i}, w_i as

$$\begin{aligned} Y_{1i} &= y_{1i}P, \\ Y_{2i} &= [(y_{2i}x_i + h_{2i}d_i)\ mod\ q]P_{pub} = (u_i, v_i), \\ w_i &= [(u_i(y_{1i} + h_{3i}x_i) + h_{4i}d_i]\ mod\ q, \end{aligned}$$

  where

$$\begin{aligned} h_{2i} &= H_2(m_i, ID_i, Y_{1i}), \\ h_{3i} &= H_3(m_i, ID_i, PK_i, t_i), \\ h_{4i} &= H_4(m_i, ID_i, PK_i, t_i). \end{aligned}$$

  - Output $\sigma_i = (Y_{1i}, u_i, w_i)$ as a signature on (m_i, t_i) for $\{ID_i, PK_i\}$.
- **Single Signature Verification:** Given a signature $\sigma_i = (Y_{1i}, u_i, w_i)$ on (m_i, t_i) for $\{ID_i, PK_i\}$, the corresponding Cluster Head (CH) checks its validity as:
  - Calculate

$$\begin{aligned} h_{1i} &= H_1(ID_i, R_i, P_{pub}), \\ h_{3i} &= H_3(m_i, ID_i, PK_i, t_i), \\ h_{4i} &= H_4(m_i, ID_i, PK_i, t_i). \end{aligned}$$

  - Check the equation

$$w_iP - u_i(Y_{i1} + h_{3i}X_i) = h_{4i}(R_i + h_{1i}P_{pub}).$$

  If it holds, accepts σ_i, otherwise, reject it.
- **Aggregate:** For an aggregate set of n distinct signatures $\{\sigma_i = (Y_{1i}, u_i, w_i)\}_{i=1}^n$ on a set of messages and timestamps $\{m_i, t_i\}_{i=1}^n$ from different sensor nodes with a set of the identities and public keys $\{ID_i, PK_i\}_{i=1}^n$, the CH generates an aggregate signatures σ as:
  - Calculate $h_{3i} = H_3(m_i, ID_i, PK_i, t_i)$ and

$$Y = \sum_{i=1}^n u_iY_{1i}, \quad U = \sum_{i=1}^n u_ih_{3i}X_i, \quad w = \sum_{i=1}^n w_i.$$

  - Output $\sigma = (Y, U, w)$ as an aggregate signature of $\{\sigma_i\}_{i=1}^n$.
- **Aggregate Verification:** For an aggregate signature σ on $\{m_i, t_i, ID_i, PK_i\}_{i=1}^n$, the MS verifies σ as:
  - Calculate

$$\begin{aligned} h_{1i} &= H_1(ID_i, R_i, P_{pub}), \\ h_{4i} &= H_4(m_i, ID_i, PK_i, t_i). \end{aligned}$$

  and check the equality

$$wP - Y - U = \sum_{i=1}^n h_{4i}(R_i + h_{1i}P_{pub}).$$

  If it holds, output valid.

III. CRYPTANALYSIS OF TWO SIGNATURE SCHEMES

Now, we present Type I and Type II attacks on Yeh's CLS scheme. We then show that Gayathri *et al.*'s CLAS scheme is insecure against universal forgery attacks.

A. ATTACKS ON YEH's CLS SCHEME

There are two types of adversaries, a Type I adversary and a Type II adversary in CLS schemes depending on their abilities.

- A Type I adversary is a malicious third party who can replace a user public key, so he/she knows a user secret value corresponding to the user public key.
- A Type II adversary is a malicious KGC who knows the master secret, but cannot know users' secret values.

In [13], the author proved that Yeh's scheme was unforgeable against both Type I and Type II adversaries under the intractability of the mathematical problem. Now, we show that Yeh's CLS scheme is still insecure against Type I and Type II attacks despite its security proofs.

■ Type I Attacks on Yeh's CLS Scheme.

- Suppose that a Type I adversary $\mathcal{A}_I$ intends to forge a certificateless signature on a message m for a user with an identity ID_i by replacing a user's public key PK_i with a new public key PK_i' its own choice.
- First, $\mathcal{A}_I$ chooses $\alpha, \beta, t_i' \in_R \mathbb{Z}_n^*$ and computes

  $$\begin{aligned} R_i' &= \beta \cdot P, \quad h_i' = H(ID_i, R_i', PK_{KGC}), \\ PK_i' &= \alpha \cdot P - h_i' \cdot PK_{KGC}, \quad T_i' = t_i'P, \\ k_i' &= H(m, T_i', PK_i', h_i'), \\ \tau_i' &= t_i' + k_i' \cdot (\alpha + ID_i \cdot \beta) \ mod\ n. \end{aligned}$$

  Then $\sigma_i' = (R_i', T_i', \tau_i')$ is a valid certificateless signature on m' for $\{ID_i, PK_i'\}$ since it satisfies the following verification equation

  $$\tau_i' \cdot P = T_i' + k_i' \cdot (PK_i' + ID_i \cdot R_i' + h_i' \cdot PK_{KGC})$$

  since

  $$\begin{aligned} &T_i' + k_i' \cdot (PK_i' + ID_i \cdot R_i' + h_i' \cdot PK_{KGC}) \\ &= T_i' + k_i' \cdot (\alpha \cdot P - h_i' \cdot PK_{KGC} \\ &\quad + ID_i \cdot \beta \cdot P + h_i' \cdot PK_{KGC}) \\ &= T_i' + k_i' \cdot (\alpha P + ID_i \cdot \beta \cdot P) \\ &= [t_i' + k_i' \cdot (\alpha + ID_i \cdot \beta)] \cdot P = \tau_i' \cdot P. \end{aligned}$$

- Finally, $\mathcal{A}_I$ succeeds in forging a certificateless signature on any message m' for $\{ID_i, PK_i'\}$ without knowing the partial private key for ID_i generated by the master secret key s. Thus, $\mathcal{A}_I$ can generate valid signatures of any messages for $\{ID_i, PK_i'\}$ at any time.

■ Type II Attacks on Yeh's CLS Scheme.

- Suppose that a Type II adversary $\mathcal{A}_{II}$, who knows the master secret s, intends to forge a signature on any message m for a user with a user public key PK_i and an identity ID_i.
- First, $\mathcal{A}_{II}$ selects $a, t_i \in_R \mathbb{Z}_n^*$ and calculates

  $$\begin{aligned} R_i &= ID_i^{-1} \cdot (aP - PK_i), \\ h_i &= H(ID_i, R_i, PK_{KGC}), \\ T_i &= tP, \quad k_i = H(m, T_i, PK_i, h_i), \\ \tau_i &= t_i + k_i \cdot (a + h_i \cdot s) \ mod\ n. \end{aligned}$$

  Then $\sigma_i = (R_i, T_i, \tau_i)$ is a valid certificateless signature for the user with $\{ID_i, PK_i\}$ since it passes the verification equation

  $$\tau_i \cdot P = T_i + k_i \cdot (PK_i + ID_i \cdot R_i + h_i \cdot PK_{KGC})$$

  since

  $$\begin{aligned} &T_i + k_i \cdot (PK_i + ID_i \cdot R_i + h_i \cdot PK_{KGC}) \\ &= T_i + k_i \cdot [PK_i + ID_i \cdot ID_i^{-1} \\ &\quad \cdot (aP - PK_i) + h_i \cdot PK_{KGC}] \\ &= T_i + k_i \cdot (aP + h_i \cdot PK_{KGC}) \\ &= [t + k_i \cdot (a + h_i \cdot s)] \cdot P = \tau_i \cdot P. \end{aligned}$$

- Finally, $\mathcal{A}_{II}$ succeeds in forging a certificateless signature on any message m for $\{ID_i, PK_i\}$ without knowing the user secret key x_i related to PK_i. Thus, $\mathcal{A}_{II}$ can generate valid signatures of any messages for $\{ID_i, PK_i\}$ at any time.

Using the vulnerabilities of Yeh's CLS Scheme, the adversaries can impersonate any user to the merchant server in Yeh's transaction scheme.

■ Impersonation Attacks on Yeh's Transaction Scheme. Using the vulnerabilities of Yeh's CLS Scheme, the adversaries can impersonate any user to the merchant server in Yeh's transaction scheme. The impersonation attacks on the transaction scheme can be mounted as follows:

- Let $\mathcal{A}_I$ be a Type I adversary who intends to impersonate a legitimate user ID_i in the transaction scheme.
- Suppose that the adversary $\mathcal{A}_I$ starts a new transaction in its own app. and the connection with the transaction identity ID_T is established between the user and the Android Pay Platform. After requesting a Full Wallet, $\mathcal{A}_I$ receives a signature σ_1 on the Full Wallet, FW. Then $\mathcal{A}_I$ checks the validity of σ_1.
- Next, $\mathcal{A}_I$ forges a signature σ_2 on the information including Credential, ID_i and ID_T as in our Type I attack on the CLS scheme.
- After checking the validity of σ_2, the merchant server believes that the transaction is completed successfully. Finally, the adversary succeeds in impersonating the user with the identity ID_i to the merchant server.

Forging valid certificateless signatures on the given messages, our impersonation attack is possible since the scheme has no authentication procedures except using the certificateless signature. The Type II adversary can also impersonate any legitimate users to the merchant server in the same way.

B. UNIVERSAL FORGERY ATTACKS ON GAYATHRI *et al.*'s SCHEMES

Now, we show that anyone can forge certificateless signatures and aggregate signatures of Gayathri *et al.*'s schemes on any messages for any identities using only publicly known information, i.e. their CLS and CLAS schemes are vulnerable to universal forgery attacks.

■ Universal Forgery Attacks on Gayathri *et al.*'s CLS Scheme.

- Let $\mathcal{A}$ be an adversary who intends to forge a certificateless signature on any message for a user with an identity ID_i and a pubic key $PK_i = (R_i, X_i)$.
- $\mathcal{A}$ chooses $\alpha, \beta \in \mathbb{Z}_q^*$, a current timestamp t_i and computes

$$\begin{aligned} h_{1i} &= H_1(ID_i, R_i, P_{pub}),\\ h_{3i} &= H_3(m_i, ID_i, PK_i, t_i),\\ h_{4i} &= H_4(m_i, ID_i, PK_i, t_i). \end{aligned}$$

Then $\mathcal{A}$ generates $\sigma_i = (Y_{i1}, u_i, w_i)$ from publicly known information as

$$\begin{aligned} u_i &= \alpha, \quad w_i = \beta,\\ Y_{i1} &= -h_{3i}X_i - \alpha^{-1}[h_{4i}(R_i + h_{1i}P_{pub}) - \beta P]. \end{aligned}$$

- Then $\sigma_i = (Y_{i1}, u_i, w_i)$ is a valid signature on m_i for $\{ID_i, PK_i\}$: it passes the verification equation

$$w_iP - u_i(Y_{i1} + h_{3i}X_i) = h_{4i}(R_i + h_{1i}P_{pub}),$$

since

$$\begin{aligned} & w_iP - u_i(Y_{i1} + h_{3i}X_i)\\ &= \beta P - \alpha(-h_{3i}X_i - \alpha^{-1}\\ &\quad \times [h_{4i}(R_i + h_{1i}P_{pub}) - \beta P] + h_{3i}X_i)\\ &= \beta P + h_{4i}(R_i + h_{1i}P_{pub}) - \beta P\\ &= h_{4i}(R_i + h_{1i}P_{pub}). \end{aligned}$$

- Finally, $\mathcal{A}_I$ succeeds in forging a certificateless signature on any message m' for $\{ID_i, PK_i\}$ from only publicly known information.

Since a signature on any message for any identity can be forged from only known information, an aggregate signature can be forged trivially. Next attacks show that certificateless aggregate signatures can be forged on the scheme without forging each individual signature.

■ **Universal Forgery Attacks on Gayathri *et al.*'s CLAS Scheme.**

- Let $\mathcal{A}$ be an adversary who intends to forge a CLAS on any message for $\{ID_i, PK_i\}_{i=1}^n$ and $\{m_i, t_i\}_{i=1}^n$.
- $\mathcal{A}$ chooses $\alpha, \beta \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} h_{1i} &= H_1(ID_i, R_i, P_{pub}),\\ h_{4i} &= H_4(m_i, ID_i, PK_i, t_i). \end{aligned}$$

Next, $\mathcal{A}$ generates a CLAS $\sigma_i = (Y_{i1}, u_i, w_i)$ from known information as

$$w = \alpha, \; Y = \alpha P - \sum_{i=1}^{n} h_{4i}R_i,$$

$$U = -\sum_{i=1}^{n} h_{4i}h_{1i}P_{pub}.$$

- Then $\sigma_i = (Y, U, w)$ is a valid CLAS on $\{m_i, t_i\}_{i=1}^n$ for $\{ID_i, PK_i\}_{i=1}^n$: it satisfies the aggregate verification equation

$$wP - Y - U = \sum_{i=1}^{n} h_{4i}(R_i + h_{1i}P_{pub}),$$

since

$$\begin{aligned} & wP - Y - U\\ &= \alpha P - (\alpha P - \sum_{i=1}^{n} h_{4i}R_i) - (-\sum_{i=1}^{n} h_{4i}h_{1i}P_{pub})\\ &= \sum_{i=1}^{n} h_{4i}(R_i + h_{1i}P_{pub}). \end{aligned}$$

- Finally, $\mathcal{A}_I$ succeeds in forging a CLAS on any set of messages and identities using only publicly known information.

In our attacks, algebraic relations in the underlying group allow the adversary to generate specific values for removing the master secret key and the user secret key that results in forging signatures and aggregate signatures of the schemes. Therefore, the schemes are entirely broken. It is trivial that Gayathri *et al.*'s CLS and CLAS schemes are insecure against both Type I and Type II adversaries with special abilities.

IV. DISCUSSIONS AND SOME IMPROVEMENTS

We showed that Yeh's CLS scheme was insecure against both Type I and Type II attacks.

- Yeh proved that the CLS scheme was unforgeable against both Type-I and Type-II adversaries under the intractability of Elliptic Curve Discrete Logarithm problem.
- Our results mean that either their security proofs are flawed or their formal security models don't cover various forgery attacks caused by these algebraic relations. However, our attacks don't mean that the underlying mathematically hard problems are solved.
- Our attacks on the two CLS schemes use algebraic relations between signatures in the underlying groups to forge signatures on any message for any users. The attacks can be prevented by destroying their algebraic relations. The algebraic relations can be destroyed from the use of hash functions by adding the required values to their inputs.

We suggest some improvements on Yeh's CLS scheme to prevent our attacks by just modifying the input of hash functions.

■ **Improvements on Yeh's CLS scheme.**

- To prevent our attacks, the **SetPublicKey**, **Sign** and **Verify** algorithms should be modified as follows:
  - **SetPublicKey:** For *params* and x_i, a user i calculates

    $$PK_i = (X_i = x_i \cdot P, \; R_i = r_iP)$$

    as its public key.
  - **Sign:** Given *params*, a partial private key D_i, a secret value x_i, and a message m, a user i picks a random number $t_i \in_R \mathbb{Z}_n$ and calculates

    $$\begin{aligned} T_i &= t_i \cdot P, \; k_i = H(m, T_i, X_i, R_i, h_i),\\ l_i &= H(m, X_i, R_i, T_i),\\ \tau_i &= t_i + k_i \cdot (l_i \cdot x_i + s_i) \; mod \; n. \end{aligned}$$

Then $\sigma_i = (R_i, T_i, \tau_i)$ is a signature on the message m.

– **Verify:** For *params*, an identity ID_i, a user public key PK_i, a message m and a signature $\sigma_i = (R_i, T_i, \tau_i)$ on m, a verifier calculates

$$h_i = H(ID_i, R_i, PK_{KGC}),$$
$$k_i = H(m, T_i, X_i, R_i, h_i)$$
$$l_i = H(m, X_i, R_i, T_i),$$

and checks following equality

$$\tau_i \cdot P = T_i + k_i \cdot (l_i \cdot X_i + ID_i \cdot R_i + h_i \cdot PK_{KGC}).$$

If it holds then accept the signature.

- In Yeh's scheme, the value $R_i = r_i P$ is contained to related to a user's partial private key. If we can set the user public key as $PK_i = (X_i = x_i P, R_i = r_i P)$ instead of $PK_i = X_i = x_i P$ in an improved scheme, since R_i is contained in all signatures. Then, the Type II adversary who knows the master secret key cannot replace the user public key $PK_i = (X_i, R_i)$, the Type II attack to change X_i or R_i can be prevented.
- To prevent our Type I attack, one can make it impossible for the adversary to replace the public key for the elimination of PK_{KGC}. In the Type I attack, an adversary can make R_i or $PK_i = X_i$ so that the part $h_i PK_{KGC}$ related to the master secret key in the verification equation is eliminated. In improved scheme, the public key is $PK_i = (X_i, R_i)$. To succeed the attack, R_i or X_i should contain $h_i PK_{KGC}$ to remove $h_i PK_{KGC}$ in the verification equation. For the verification equation of the improved scheme, $\tau_i \cdot P = T_i + k_i \cdot (l_i \cdot X_i + ID_i \cdot R_i + h_i \cdot PK_{KGC})$, R_i cannot contain the value $h_i PK_{KGC}$ since $h_i = H(ID_i, R_i, PK_{KGC})$ contains R_i as its input. Also, X_i cannot contain the value $h_i PK_{KGC}$ since $l_i = H(m, R_i, X_i, T_i)$ has X_i as its input. It needs $h_i PK_{KGC}$ multiplied by l_i^{-1} to eliminate $h_i PK_{KGC}$ in the above equation, but it is impossible to compute X_i so that X_i should contain l_i.

We will not suggest improvements on Gayathri *et al.*'s CLS and CLAS schemes against our attacks since their schemes are entirely broken. The CLAS scheme allows to aggregate n distinct signatures on n distinct messages for n distinct users into a single signature. There have been proposed a number of CLAS schemes for various IoT applications since they can reduce computation overhead for verification and communication/storage overhead. The size of their aggregate signatures in most of CLAS schemes grows linearly with the number of signers in an aggregating set. It remains an open problem to propose a secure and efficient CLAS scheme with constant-size aggregate signatures.

V. CONCLUSION

We showed that Yeh's CLS scheme was insecure against both Type I and Type II attacks. Despite the security proofs reduced to the intractability of the Elliptic Curve Discrete Logarithm problem, we showed that the scheme was insecure against Type I and Type II using algebraic relations between signatures in the underlying groups to forge certificateless signatures. Our attacks don't mean that the underlying mathematically hard problems are solved. We also showed that Gayathri *et al.*'s CLS and CLAS schemes were vulnerable to the universal forgery attacks although their security against Type I and II adversaries were proven under the hardness of the mathematical hard problem. Our attacks mean that their security proofs are flawed or their formal security models don't cover various forgery attacks caused by these algebraic relations. Therefore, to design secure signature schemes, their exact security analysis and security proofs against algebraic attacks should be required.

REFERENCES

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT 2003* (Lecture Notes in Computer Science), vol. 2894. New York, NY, USA: Springer-Verlag, 2003, pp. 452–473.

[2] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *J. Med. Syst.*, vol. 36, no. 1, pp. 93–101, Feb. 2012.

[3] (2016). *Android Pay API Process Flow*. Accessed: May 10, 2020. [Online]. Available: https://developers.google.com/android-pay/diagrams

[4] (2016). *Android Pay API Tutorial*. Accessed: May 10, 2020. [Online]. Available: https://developers. google.com/android-pay/android/tutorial

[5] (2016). *Apple Pay*. Accessed: May 10, 2020. [Online]. Available: https://www.apple.com/apple-pay

[6] G. V. Crosby, T. Ghosh, R. Murimi, and C. A. Chin, "Wireless body area networks for healthcare: A survey," *Int. J. Ad Hoc, Sensor Ubiquitous Comput.*, vol. 3, no. 3, p. 1, 2012.

[7] (2016). *FitPay Smart Strap*. Accessed: May 10, 2020. [Online]. Available: https://www.pagare.me/

[8] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[9] N. B. Gayathri, G. Thumbur, P. Rajesh Kumar, M. Z. U. Rahman, P. V. Reddy, and A. Lay-Ekuakille, "Efficient and secure pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9064–9075, Oct. 2019.

[10] P. Kumar and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors*, vol. 12, no. 1, pp. 9–55, 2012.

[11] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196. Berlin, Germany: Springer, 1984, pp. 47–53.

[12] (2016). *Topshop Bpay Accessories*. Accessed: May 10, 2020. [Online]. Available: http://www.topshop.com/en/tsuk/category/bpay-4991797/home?geoip=noredirect

[13] K.-H. Yeh, "A secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," *IEEE Syst. J.*, vol. 12, no. 2, pp. 2027–2038, Jun. 2018.

KYUNG-AH SHIM (Member, IEEE) is currently a Senior Researcher with the National Institute for Mathematical Sciences. Her research interests include public-key cryptography, post-quantum cryptography, cryptographic protocols, and information security.