Research Article

Cryptanalysis of a Certificateless Hybrid Signcryption Scheme and a Certificateless Encryption Scheme for Internet of Things

Shan Shan

School of Economics and Management, Shandong Jiaotong University, Jinan 250357, China

Correspondence should be addressed to Shan Shan; jtsanshan@126.com

Received 9 April 2022; Revised 13 June 2022; Accepted 29 June 2022; Published 13 July 2022

1. Introduction

The primary problem to be solved in public key cryptography is how to certify the ownership of key pairs. In certificate-based public key infrastructure (PKI), a trusted third party called certificate authority (CA) issues certificates that provide a trusted link between the user’s identity and the public key based on digital signature technology. However, certificate management is very complex. Shamir [1] proposed the concept of an identity-based cryptosystem in 1984 to simplify certificate management issues. The main idea is that you can easily export a user’s public key from any string that corresponds to the user’s identifying information, such as name, phone number, and E-mail address. A private key generator (PKG) calculates the private keys using the master key and securely distributes these private keys to the users participating in the scheme. From an efficiency and convenience standpoint, an identity-based system may be a good alternative to a certificate-based system. But key escrow, which means the user’s private key is generated and known by PKG, is an inherent problem resulting in no authenticity and no privacy for the user in an identity-based system.

As a variant of the identity-based cryptosystem, the concept of certificateless was proposed in 2003 to eliminate these problems simultaneously [2]. Each user in a certificateless scheme independently generates a secret key and gets another partial private key from the key generation center (KGC). Thus, each user’s secret consists of two parts, one obtained from a trusted third party (KGC) and one generated by the user himself. Certificateless scheme successfully solves the key escrow problem. In addition, this kind of scheme does not require the trusted third party to authenticate the corresponding public key ownership, which makes public key management very efficient. Because of these advantages, certificateless schemes have attracted wide attention and become one of the hot topics of public key cryptography. In recent years, certificateless signcryption [3, 4], certificateless hybrid signcryption [5, 6], certificateless multireceiver signcryption [7–9], certificateless generalized signcryption [10–14], and certificateless online/offline signcryption [15, 16] have been put forward one after another.

In wireless and mobile networks with limited storage and computing resources, certificateless cryptography has more advantages because of its low dependence on infrastructure and short security parameters. However, while achieving low computational costs, many certificateless schemes proposed in the Internet of things environment [17–23] cannot simultaneously provide provable security. Kumar et al. [17]
claimed that their newly proposed certificateless aggregate
signature scheme is secure against both types of attackers. Zhan and Wang [24] proved that an attacker could forge a
valid signature and valid aggregate signature. Lin et al. [25]
pointed out that the certificateless signcryption (CL-SC)
scheme proposed by Rastegari et al. [18] is insecure. Zhan
et al. [26] analyzed a pairing-free CLAS scheme proposed in
[20] and pointed out that the scheme is insecure. On this
basis, to solve the security vulnerability, an improved scheme
was proposed at the same time. Khan et al. [21] proposed a
certificateless offline/online signature scheme. Unfortunately,
their scheme is not secure against adaptive selective
message attacks. Hussain et al. [27] proved that an adversary
could forge a valid signature on a message by replacing a
public key. Kasyoka et al. [28] showed the security vul-
erabilities of Wei and Ma’s [19] signcryption scheme and
proposed corresponding modifications to show how their
scheme could be made more secure. Xu and Zeng [29]
pointed out that the certificateless aggregate arbitrated
signature scheme proposed by Lee et al. [22] is not secure for
type-1 attackers that can replace user public keys. They also
showed that Addobea et al.’s [23] offline/online certificate-
less signature scheme cannot achieve correctness.
Therefore, the certificateless solution described above cannot
be deployed in real Internet of things environment and
mobile applications. Most of the schemes fail because the
definition of the security model is not complete, and in the
proving process, the adversary capability is not successfully
reduced to solve difficult problems. There has been an
ongoing effort in the Internet of things to make greater
advances in security and performance.

1.1. Our Contributions. Recently, Gong et al. [30] and Karati
et al. [31], respectively, proposed a new certificateless scheme
in the Internet of things environment, one of which is a
certificateless hybrid signcryption scheme, and the other’s
basis is a certificateless encryption scheme. Their schemes
were claimed to be secure, and the formal security was
presented which reducing adversary capabilities in solving
difficult problems. It is a pity that Gong et al.’s scheme and
Karati et al.’s scheme are not secure in the case of internal
difficult problems. It is a pity that Gong et al.’s scheme and
Karati et al.’s scheme are not secure in the case of internal
attacks as shown in this paper. The attack algorithms against
these two schemes are presented separately, thus proving
that their schemes are insecure and not suitable for the
Internet of things environment.

1.2. Paper Organization. In Section 2, we give the crypt-
analysis of Gong et al.’s scheme, and we give the crypt-
analysis of Karati et al.’s certificateless encryption scheme for
the industrial Internet of things in Section 3. Section 4
provides a conclusion.

2. Cryptanalysis of Gong et al.’s Certificateless
Hybrid Signcryption Scheme

Because of the limitation of symmetric cryptography, public
key-based authentication technology has attracted extensive
attention. It provides secure communication and accesses
mechanism for various applications. Compared with single-
factor or two-factor protocols, multifactor schemes have been
proven to achieve higher security levels. Wang et al.
[32–34] have made a series of representative achievements in
multifactor authentication. However, in some applications,
people have to strike a balance between availability and
security and adopt single-factor technology to achieve au-
thentication, such as digital signature and digital sign-
cryption. Signcryption can provide confidentiality and
authentication at the same time and is widely used in many
applications where multiple security features are required.
Gong et al.’s scheme is a concrete certificateless hybrid
signcryption scheme.

2.1. Gong et al.’s Scheme. As shown below, their scheme
includes five algorithms altogether: Setup, Extract-Partial-
Private-Key, Generate-User-Keys, Signcryption, and
Unsigncryption.

2.1.1. Setup. KGC runs the following algorithms:

(i) Generate two distinct cyclic groups $G_1$ (an additive
cyclic group with a generator $P$) and $G_2$ (a multi-
plicative cyclic group) of prime order $q (q \geq 2^r)$. $e$ is
a bilinear map.

(ii) Chooses $x \in \mathbb{Z}_q^*$, computes $P_{pub} = e(P, P)^x$.

(iii) Chooses one-way hash functions as $h_1: \{0, 1\}^n \rightarrow
\mathbb{Z}_q^*$, $h_2: \{0, 1\}^{n+1} \times G_1 \times G_2^* \rightarrow \mathbb{Z}_q^*$,
h_3: \mathbb{Z}_q^* \rightarrow \mathbb{Z}_q^*$.

(iv) Finally, keeps $x$ safely and outputs params = {$P_
_{pub}, G_1, G_2, q, e, n, h_1, E, D$} as the system
parameter.

2.1.2. Extract-Partial-Private-Key. Given the identity in-
formation $u_i$, to generate the corresponding partial private
dkey $d_{i}$, KGC runs the following algorithms:

(i) Computes $Q_{i} = h_1(u_i)$.

(ii) Sets the partial private key $d_{i} = xh_1(u_i)$

2.1.3. Generate-User-Keys. The user chooses $x_i \in \mathbb{Z}_q^*$
and computes $P_i = e(P, P)^x_i$, which is the public key and sets the
full private key $s_i = (x_i, d_i)$.

2.1.4. Signcryption. A sender $u_A$ runs the following algorithms
to generate the ciphertext.

(i) Chooses $r \in \mathbb{Z}_q^*$

(ii) Computes $R = rP$, $y = P_B^{x_i}h_2(r)$ and $z =
h_1(Q_B \cdot d_B)$, where $Q_B = h_1(u_B)$

(iii) Computes $K = h_1(y, z, R)$ and $c = Enc_K(m)$

(iv) Computes $f = h_2(u_A, u_B, R, P_A, P_B)$ and $s =
r \cdot z / x_A \cdot f$

(v) Outputs $\sigma = (c, R, s)$ as the ciphertext.
2.1.5. Unsigncrypt. A receiver $u_B$ runs the following algorithms for unsigncation.

(i) Computes $y = P^{x h_3(y)} R$, $z = h_3(Q_A \cdot d_B)$, $Q_A = h_1(u_A)$ and $K = h_1(y, z, R)$.
(ii) Computes message $m = \text{Dec}_K(c)$. If output $\perp$, $u_B$ refuses the message.
(iii) Computes $f = h_3(u_A, u_B, R, P_A, P_B)$.
(iv) Checks $P_A^{x f} = e(zP, R)$ holds or not. If it holds, $u_B$ get $m$; else $u_B$ refuses the message.

2.2. Cryptanalysis of Gong et al.’s Scheme

2.2.1. Attack Algorithm 1 (Internal Attacks to the Unforgeability). Once receives a valid signcryption text $\sigma = (c, R, s)$, the receiver can impersonate the sender to generate signcryption text for any message $m'$ sent to him. The attack algorithm is described as follows:

(i) Chooses $r' \in \mathbb{Z}_q$.
(ii) Computes $R' = r' R \cdot R, z = h_3(Q_A \cdot d_B)$, and $f = h_2(u_A, u_B, R, P_A, P_B)$.
(iii) Computes $y' = P^{x h_3(R')} A, z' = h_3(Q_A \cdot d_B)$, where $Q_A = h_1(u_A)$.
(iv) Computes $K' = h_4(y', z', R')$.
(v) Computes $c' = \text{Enc}_{K'} (m')$.
(vi) Computes $f' = h_5(u_A, u_B, R, P_A, P_B)$ and $s' = s \cdot f \cdot z' / f'$. 
(vii) Send the ciphertext $\sigma' = (c', R', s')$ of message $m'$.

2.2.2. Correctness. The signcryption ciphertext $\sigma' = (c', R', s')$ is validly related with $m'$ as shown in the following.

Since $R' = r' R \cdot R, y' = P^{x h_3(R')} A = P^{x h_3(R')} A$, $z' = h_3(Q_A \cdot d_B)$, the receiver can compute $m' = \text{Dec}_{K'} (c')$ where $K' = h_4(y', z', R')$.

The equation $P_A^{x f'} = e(zP, R')$ always holds since

\[
\begin{align*}
\sigma' &= (c', R', s') \\
&= (c, R, s) \cdot f' / f \\
&= (r \cdot z', r' \cdot z', f' / f) \cdot (A \cdot f) \\
&= (r \cdot z', r' \cdot z', z' / f).
\end{align*}
\]

Thus, $\sigma' = (c', R', s')$ is a valid signcryption ciphertext.

Any user can launch the attack after receiving a valid signcryption ciphertext sent to him, so the nonrepudiation and source authentication that should be satisfied by the digital signcryption scheme cannot be realized.

2.2.3. Attack Algorithm 2 (Internal Attacks to the Master Secret Key). As shown in the Extract-Partial-Private-Key algorithm, KGC generates $d_i$ by computing $Q_i = h_1(u_i)$ and $d_i = x h_1(u_i)$.

Since $x$ is a random element in $\mathbb{Z}_q$ and $h_1$ is a hash function that maps strings to distinct elements in $\mathbb{Z}_q$, any partial private key holder can compute the master secret key $x$ by $x = d_i \cdot h_1^{-1}(u_i) \in \mathbb{Z}_q$ directly. Any security of the whole system cannot be realized when the master secret key is leaked. Any user that receives a valid partial private key can launch the attack.

3. Cryptanalysis of Karati et al.’s Certificateless Encryption Scheme

In order to achieve more complex security goals, people often adopt the method of extending features on the basis of the general scheme. Karati et al.’s reliable data sharing protocol is based on a certificateless encryption scheme.

3.1. Karati et al.’s Scheme. As shown below, their scheme includes ten algorithms: Setup, Set-Secret-Value, Set-Public-Value, Set-Partial-Private-Key, Set-Full-Private-Key, Encrypt, Gen-Trapdoor, Test-Trapdoor, and Decrypt.

3.1.1. Setup. KGC runs the following algorithms.

(i) Generates three distinct cyclic groups $G_1, G_2$, and $G_3$, and $e: G_1 \times G_2 \rightarrow G_3$ is a bilinear map.
(ii) Chooses generator $g \in G_1, h \in G_2$.
(iii) Chooses $H_1: \{0, 1\}^n \times G_1 \times G_2 \times G_4 \rightarrow \mathbb{Z}_{p}$, $H_2: \{0, 1\}^n \rightarrow \mathbb{Z}_{p}$, and $H_3: G_3 \rightarrow \{0, 1\}^{n_1 \cdot n_2}$ for some $n_1$ and $n_2$, which are one-way hash functions.
(iv) Computes $g_1 = g_1$ for $x_{KGC} \in \mathbb{Z}_{p}$.
(v) Keeps $MSK = (x_{KGC})$ safely and publishes $\text{params} = \{ p, g, g_1, h, H_1 \}$.

3.1.2. Set-Secret-Value and Set-Public-Value.

(i) Chooses $y_i \in \mathbb{Z}_p$ and sets secret-value $SS_i = (y_i)$.
(ii) Generates public value $PV_i = (P_{11} = h^{y_i}, P_{12} = e(g, h)^{y_i})$.

3.1.3. Set-Partial-Private-Key. KGC runs the following algorithms to generate the partial private key of device $i$:

(i) Chooses $\beta_i \in \mathbb{Z}_p$ and $d_i \in \mathbb{Z}_p$.
(ii) Computes $P_{13} = g^{\beta_i}$ and $a_i = H_1(ID_p, P_{13}, P_{11}, P_{12})$.
(iii) Computes $x_i = 1 / (a_i \beta_i + d_i x_{KGC})$ and $D_i = h^{x_i}$.
3.1.4. Set-Full-Public-Key and Set-Full-Private-Key. The full public key of Device $i$ can be expressed as $PK_i = (P_i, P_{i3}, P_j)$, and the full private key can be expressed as $SK_i = (y_{ij}, d_i, D_i = D_{i1/2y})$.

3.1.5. Encrypt. Given the message $m_i$ and keyword $w_{ij} \in \{0, 1\}^{n_i}$, a sender, whose private key is $SK_i$, runs the following algorithms to generate a ciphertext sending to receiver $R$ with public key $PK_r$.

(i) Choose $u \in \mathbb{Z}_p^*$ and $\sigma \in \mathbb{R}[0, 1]^{\mathbb{R}}$.

(ii) Sets $C_i = (c_{ij} = g^{nu_i}, c_{ij} = g^{nu_i}, c_{ij} = (m_i \| \sigma) \oplus H_5(P_{ij}))$.

(iii) Computes $v = H_1((s \| w_{ij})y_j, g, h, P_{i2})$, $\Phi_{ij} = (\phi_{ij1}, \phi_{ij2})$, where $\phi_{ij1} = [g^i \cdot g^{H_2(w_{ij}\|s\|\beta_i)}]$ and $\phi_{ij2} = P_{ij}^{\gamma_i}$.

(iv) Outputs $(C_i, \Phi_{ij}, \theta_{ij})$ for $\theta_{ij} = H_2(m_i \| s \| \beta_i)$.

3.1.6. Gen-Trapdoor. Given a tester’s public-private key pair $(\epsilon \in \mathbb{Z}_p^*, (P_{ij} = h^\epsilon, P_{ij} = e(g, h)^{1/2y_j}))$, receiver $R$ runs the following algorithms to generate a trapdoor $\Gamma_{ik} = (\tau_{ik1}, \tau_{ik2})$.

(i) Computes $v' = H_1((s \| w_{ij})y_j, g, h, P_{i2})$. 

(ii) Computes $\tau_{ik1} = [g_i \cdot g^{H_2(w_{ij}\|s\|\beta_i)}]$ and $\tau_{ik2} = P_{ij}^{\gamma_i/y_i,H_1(s, g, P_{ij}^\gamma_i, P_{i2})}$ for $\beta_i \in \{0, 1\}^{n_i}$.

3.1.7. Test-Trapdoor. The tester computes $v'' = H_1(\alpha_i, g, P_{j1}, P_{j2})$ and retrieves $C_i$ if the condition $e(\phi_{ij1}, \tau_{ik1}) \cdot \gamma_i = e(\tau_{ik1}, \phi_{ij2})$ holds.

3.1.8. Decrypt. Given a keyword $w_{ik}$, $SK_r = (y_j, d_i, D_i)$, $C_i$, the receiver computes $\delta_i = e(c_{ij}^{\gamma_i}, D_i)$ and $\delta_i = (m_i \| \sigma) = c_{ij} \oplus H_5(e)$. The first $n_i$ bit of $\delta_i$ is returned as $m_i$ if $\theta_{ij} = H_2(\delta_i \| w_{ik})$.

3.2. Cryptanalysis of Karati et al.‘s Scheme. To show the usability, Karati et al. defined their scheme as $(M, C, W, \Gamma)$-KDCLEKS. We noticed that if the sender sends a message directly without any keyword, $(M, C, \Gamma)$-KDCLEKS is a common certificateless encryption scheme, which can be marked as $(M, C)$-KDCLEKS.

In this section, it will be shown that the encryption algorithm $(M, C)$-KDCLEKS is not secure under public-key replacement attacks launched by an adversary $\Lambda_i$.

3.2.1. Attack Algorithm 1 (Internal Attacks to the Partial Private Key). Assume the following conditions a user declares his public value as $PV_J = (P_{j1} = h^{y_j}, P_{j2} = e(g, h)^{1/2y_j})$. Once $\Lambda_j$ receives a valid partial private key $PP_j = (d_j, P_{j3}, D_j)$, it can calculate and generate a partial private key for this user as follows:

1. Compute $P_{j3} = P_{j1}^{y_j}$ and $\alpha_j = H_1(ID_j, P_{j3}, P_{j1}, P_{j2})$.
2. Compute $d_j = \gamma_j \cdot d_i$.
3. Compute $D_j' = D_{i1/2y}^{1/2y_j}$.

3.2.2. Correctness. $PP_j = (d_j, P_{j3}, D_j')$ is a valid partial private key related to public value $PV_j$ as shown in the following equation:

$$e\left(\frac{P_{j1}^{y_j}}{P_{j1}^{\frac{y_j}{2}}}, D_j'\right) = e\left(\frac{P_{j1}^{y_j}}{P_{j1}^{\frac{y_j}{2}}}, D_{i1/2y}^{1/2y_j}\right)$$

$$= e\left(g^{\frac{y_j}{2}}, D_{i1/2y}^{1/2y_j}\right)$$

Thus, $PP_j = (d_j, P_{j3}, D_j')$ can always be accepted as a valid partial private key related to public value $PV_j$. Any user that receives a valid partial private key can launch the attack. This means that the user’s partial key can be forged, leading to the lack of availability.

3.2.3. Attack Algorithm 2 (Internal Attacks to the Confidentiality). Once $\Lambda_i$ receives a valid Full-Public-Key $PK_i = (P_{i1}, P_{i2}, P_{i3})$ and corresponding Full-Private-Key $SK_i = (y_i, d_i, D_i)$, he can decrypt the ciphertext of any user $J$ with $ID_j$ through public key replacement attacks. The attack algorithm is described as follows:

1. Select random parameter $y_j', y_j \in \mathbb{Z}_p^*$ and compute $P_{j1} = h^{y_j'}$, $P_{j2} = e(g, h)^{1/2y_j'}$ and $P_{j3} = P_{j1}^{y_j'}$. Any user $J$ with $ID_j$ can successfully decrypt it.

2. Replace the public key of user $J$ with the value $PK_j = (P_{j1}, P_{j2}, P_{j3})$.

On inputs $params$ and receiver $J$’s public key $PK_j$ with message $m_i \in \{0, 1\}^{n_i}$, the sender selects $\sigma \in \mathbb{Z}_p^*$ and $sets \epsilon \in \mathbb{Z}_p^*$ and sets $C_i = (c_{ij}, c_{ij}, c_{ij})$, where $c_{ij} = P_{ij}^{y_j}$, $c_{ij} = g^{\frac{y_j}{2}}$, $c_{ij} = (m_i \| \sigma) \oplus H_5(P_{ij})$ where $\alpha_j = H_1(ID_j, P_{j3}, P_{j1}, P_{j2})$. Finally, the sender outputs $C_i$ as the ciphertext.

Given the ciphertext $C_i$, $\Lambda_i$ can successfully decrypt it using the following algorithm:

1. Compute $\alpha_j = H_1(ID_j, P_{j3}, P_{j1}, P_{j2})$.
2. Compute $y_j = y_j' - y_j$, $d_j = \alpha_j \cdot d_i$, $D_j = D_j^{1/2y_j}$.
3. Compute $\delta_i = (m_i \| \sigma) = c_{ij} \oplus H_5 (\delta_i)$, where $\delta_i = e(c_{ij}, c_{ij}, D_j)$.

3.2.4. Correctness. The decryption process is always successful as shown in the following equation:
\[ \delta_2 = e^\left( c_j \mid d_j \right) \\
= e^{\left( p_{13}^\mu \cdot y_{13}^\nu, D_1^\nu \left( v, y \right) \right)} \] \\
= e^{\left( g^\mu \left( h^\nu \cdot y \right), h^{\nu v} \left( v, y \right) \right)} \\
= e^{\left( g^\mu \left( h^\nu \cdot y \right), h^{\nu v} \left( v, y \right) \right)} (4) \\
= e^{\left( g, h \right)^{\nu v} \left( y, y \right)} = p_{12}^\nu.

Thus, \( \mathcal{A}_j \) reveals \( m = c_j @ H_3 (\delta_2) \) with probability 1. This attack can be launched by a user who receives any legal partial private key sent to him, and he can decrypt the ciphertext of any user through public key replacement attacks without knowing the master secret MSK. This means that any user’s public key can be replaced, and the message can be revealed by the attacker, leading to the lack of confidentiality.

4. Conclusion

Gong et al. gave a formal security proof in the random oracle model, and Karati et al. proved their scheme is secure against adversaries. Unfortunately, we noticed that in Gong et al.’s scheme, internal users can forge the signcryption ciphertext sent to them, the nonreputation and source authentication that should be satisfied by the digital signcryption scheme cannot be realized. The more serious is that any partial private key holder can directly calculate the master secret key, which leads to the failure to implement security features. Any user who obtains a partial private key in Karati et al.’s basic certificateless encryption scheme can either forge the partial private key of another user or replace the public key of another user to decrypt the ciphertext. Therefore, their solutions are insecure and not suitable for the Internet of things environment.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References


