CRYPTANALYSIS OF EFFICIENT CERTIFICATELESS AGGREGATE SIGNATURE SCHEME

PANKAJ KUMAR, VISHNU SHARMA, AND VINOD KUMAR

ABSTRACT. Certificateless signature scheme becomes a most attractive area for researchers to provide a lot of potential to secure network against malicious adversaries. Aggregate signature is a many to one map that allows mapping n users signature in to single short signature. Aggregate signatures reduce the bandwidth used in the network which is useful in many practical areas where bandwidth is a major issue like vehicular ad-hoc network, PDA, wireless sensor network and an endless list. Recently, Deng et al.’s proposed a secure certificate aggregate signature scheme and claims that their scheme is secure against malicious adversaries present in the network. Unfortunately, we found that their proposed scheme is not secure against collision resistance attack. In this paper, we demonstrate that Deng et al.’s proposed scheme is fails to protect collision resistant property.

1. Introduction

In public key cryptography, Digital signature is a key feature that assures authenticity, integrity and nonrepudiation in the network. Shamir [1] proposed an Identity Based public key cryptography (ID-PKC) signature scheme that solve the certification problem arise in public key cryptography and has no need of certification for binding private/public key pair. In ID-PKC, user chose its public key such as their driving license, phone number, address or any other unique identity. Third party say, private key generator (PKG) involve in ID-PKC, generates the private key of user. Although we assume PKG as a trusted party but one possibility arise, if PKG became malicious who is responsible for generating the private key of user creates key escrow problem. Key escrow problem defines, if PKG, who is responsible for generating private key of user becomes malicious that can break the security very easily with the help of private key. In 2003, Al Riyami and Peterson [2] recommend a solution to solve Key escrow problem inherit in ID-PKC. Al Riyami and Peterson [2] proposed a certificateless signature scheme (CLS), in which key generation center (KGC) is the third party which produces the partial private key of user instead of private key and private key is generated by the user with the help of partial private key. In this case KGC don’t have knowledge of private key directly. Boneh [3] introduce the concept of aggregate signature scheme in Eurocrypt 2003. In aggregate signature scheme, aggregator collects all individual n signatures and aggregates them to produce a compact signature. Aggregate

2000 Mathematics Subject Classification. Primary 11T71; Secondary 14B50.
Key words and phrases. Certificateless signature scheme, Certificateless aggregate signature scheme, Collision resistant attack.
signatures are very beneficial in practical real life application where bandwidth limitation create a big issue such as ad-hoc networks, wireless sensor networks, internet of things and an endless list. Certificateless aggregate signatures takes the advantage of certificateless and aggregation models, it generates the solution of certificate problem and decreases the computations and restrict the bandwidth.


Recently Deng et al. [11] proposed an efficient CL-AS scheme unfortunately we found that their CL-AS scheme is not secure against collision resistance attack. In this paper we demonstrate how Deng et al. [11] CL-AS scheme fails to protect against collision resistance attack.

1.1. Collision resistance attack. Collision resistant property defined that no signer groups holding the KGC together can generate a valid aggregate signature [6]. A dishonest user might be an internal user (one of the sharing user in the aggregate signature) or external user cooperates with malicious KGC to produce a valid forge aggregate signature.

![Figure 1](image-url)


<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>The master key of KGC</td>
</tr>
<tr>
<td>$P$</td>
<td>Generator of the group</td>
</tr>
<tr>
<td>$P_{pub}$</td>
<td>The public key of KGC</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>The User’s identity</td>
</tr>
<tr>
<td>$usk_{ID_i}$</td>
<td>The Users secret key</td>
</tr>
<tr>
<td>$psk_{ID_i}$</td>
<td>The partial private key of identity $ID_i$</td>
</tr>
<tr>
<td>$Params$</td>
<td>The system parameters generated by KGC</td>
</tr>
<tr>
<td>$(usk_{ID_i}, upk_{ID_i})$</td>
<td>The user’s secret/public key pair of identity $ID_i$</td>
</tr>
<tr>
<td>$upk_{ID_i}$</td>
<td>The public key of vehicle having identity $ID_i$</td>
</tr>
<tr>
<td>$m_i$</td>
<td>The message corresponding to user’s identity $ID_i$</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>Signature on the message $m_i$ with user’s identity $ID_i$</td>
</tr>
<tr>
<td>$V$</td>
<td>Signature of user corresponding to user’s $ID_i$</td>
</tr>
<tr>
<td>KGC</td>
<td>Key Generation Center</td>
</tr>
<tr>
<td>CLS</td>
<td>Certificateless Signature</td>
</tr>
<tr>
<td>CLAS</td>
<td>Certificateless Aggregate Signature</td>
</tr>
</tbody>
</table>

**Master-key-gen:** KGC runs the algorithm after taking an input security parameter $k$.

Generate two cyclic groups $G_1$ and $G_2$, where $G_1$ additive group and $G_2$ is multiplicative group with the same order $q$ with two generator $P, Q$ of $G_1$ and a bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$

(i) Select a random number $s \in \mathbb{Z}_q^*$ and computes $P_{pub} = sP$, taking $s$ as a master key of KGC and $P_{pub}$ as a public key of KGC.

(ii) Select five one way cryptography hash functions $H_1 : \{0, 1\}^* \rightarrow G_1, H_2 : \{0, 1\}^* \rightarrow G_1, H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*, H_4 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*, H_5 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.

(iii) Generates the system parameters say $Params$ are $\{q, G_1, G_2, e, P, Q, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$ and keep secretly master key $s$ by KGC.

**Partial-Private-Key-Gen:** After taking input user’s identity $ID_i$, The KGC first computes the user’s partial private key $psk_{ID_i} = sQ_{ID_i}$ where $Q_{ID_i} = H_1(ID_i)$ and forward it to the user via a secure way.

**User-Key-Gen:** The user chooses a random number $x_{ID_i} \in \mathbb{Z}_q^*$ and set as secret key $usk_{ID_i}$, then computes its public key $upk_{ID_i} = usk_{ID_i} \cdot P$. 

Finally, section 4 and section 5 describe the conclusion and references.
Sign: The user with identity $ID_i$ takes the \emph{Params}, the partial private key $psk_{ID_i}$, corresponding secret key $usk_{ID_i}$ and then performs the following steps to generate the signature:

(i) Select a random number $r_i \in \mathbb{Z}_q^*$ and computes
\[
U_i = r_i \cdot P, \quad t_i = H_3(m_i, ID_i, upk_{ID_i}, U_i), \quad h_i = H_4(m_i, ID_i, upk_{ID_i}, U_i), \quad Q = H_2(q, P, P_{pub}), \quad k_i = H_5(m_i, ID_i, upk_{ID_i}, U_i)
\]

(ii) Compute: $V_i = psk_{ID_i} + t_i \cdot r_i \cdot P_{pub} + h_i \cdot x_{ID_i} \cdot Q + k_i \cdot r_i \cdot Q$

(iii) Provides a signature $(U_i, V_i)$ on message $m_i$.

Verify: Given a signature $(U_i, V_i)$ with message $m$ corresponding public key $upk_{ID_i}$ regarding the identity $ID_i$, verifier performs the following steps:

(i) Computes $U_i = r_i \cdot P, t_i = H_3(m_i, ID_i, upk_{ID_i}, U_i), h_i = H_4(m_i, ID_i, upk_{ID_i}, U_i), Q = H_2(q, P, P_{pub}), k_i = H_5(m_i, ID_i, upk_{ID_i}, U_i)$

Verify the following equation $e(V_i, P) = e(Q_{ID_i} + t_iU_i, P_{pub}) \cdot e(h_iupk_{ID_i} + k_iU_i, Q)$

If it satisfied then accept the signature.

2.1. Deng \emph{et al.} \cite{Deng2001} CL-AS scheme. CLAS scheme consist of seven steps in which five algorithms \emph{Master-key-gen}, \emph{Partial-key-gen}, \emph{User-key-gen}, \emph{Sign}, \emph{Verify} are same as CLS scheme and two extra algorithms say \emph{Aggregate} and \emph{AggregateVerify} are involved in CLAS scheme whose description is given below:

(1) \emph{Aggregate}: For an aggregating set of $n$ users $\{U_1, U_2, \ldots, U_n\}$ with their identities $\{ID_1, ID_2, \ldots, ID_n\}$ and the corresponding public keys $\{upk_{ID_1}, upk_{ID_2}, \ldots, upk_{ID_n}\}$, and with signature pairs $\{(m_1, \sigma_1 = (U_1, V_1)), \ldots, (m_n, \sigma_n = (U_n, V_n))\}$, then aggregator computes $V = \sum_{i=1}^{n} V_i$ and results an aggregate signature as $\sigma = (U_1, U_2, \ldots, U_n, V)$.

(2) \emph{AggregateVerify}: For verify an aggregate signature $\sigma = (U_1, U_2, \ldots, U_n, V)$ signing by $n$ users $\{U_1, U_2, \ldots, U_n\}$ with their identities $\{ID_1, ID_2, \ldots, ID_n\}$, verifier performs the following steps:

(i) Computes $Q_{ID_i} = H_3(ID_i), t_i = H_3(m_i, ID_i, upk_{ID_i}, U_i), h_i = H_4(m_i, ID_i, upk_{ID_i}, U_i), k_i = H_5(m_i, ID_i, upk_{ID_i}, U_i), Q = H_2(q, P, P_{pub})$.

(ii) Verify $e(V, P) = e(\sum_{i=1}^{n} (Q_{ID_i} + t_iU_i), P_{pub}) \cdot e(\sum_{i=1}^{n} (h_iupk_{ID_i} + k_iU_i), Q)$.

3. Cryptanalysis of Deng \emph{et al.} \cite{Deng2001} Certificate Aggregate Signature Scheme

Description of collision resistance attack is same as describe in the previous section 1.1.

KGC collaborate with the dishonest user $u_{n+1}$ of identity $ID_{n+1}$ to forge a certificateless aggregate signature scheme. This attack will be performing in 3 steps.

Step 1. A dishonest signer $u_{n+1}$ chooses a random number $v \in \mathbb{Z}_q^*$, such that $v = usk_{ID_1} + usk_{ID_2} + \ldots + usk_{ID_n} + \gamma$ afterward $u_{n+1}$ can calculate $\gamma P = vP - \sum_{i=1}^{n} upk_{ID_i}$ and put its public key while $u_n$ don’t have a knowledge about $\gamma$. 
Step 2. KGC and $u_{n+1}$ chooses $r_1, r_2, \ldots, r_n, r_{n+1} \in \mathbb{Z}_q^*$ in cooperation and calculate $U_i = r_i P$ and $k_i = H_3(m_i, u_{pkID_i}, ID_i, U_i)$, $t_i = H_2(m_i, upk_{ID_i}, ID_i, U_i)$, $h_i = H_4(m_i, upk_{ID_i}, ID_i, U_i)$, where $i \in [1, n]$. Then compute $V^* = \sum_{i=1}^{n+1} psk_{ID_i} + \sum_{i=1}^{n+1} t_i \cdot r_i \cdot P_{pub} + \sum_{i=1}^{n+1} h_i \cdot v \cdot Q + \sum_{i=1}^{n+1} k_i \cdot r_i \cdot Q$. Since all $psk_{ID_i}$ are well-known to malicious KGC afterwards they generates an aggregate signature $\sigma^* = (U_1, U_2, \ldots, U_{n+1}, V^*)$ without using the private key of corresponding identities $\{ID_1, ID_2, \ldots, ID_{n+1}\}$.

Step 3. We demonstrate that $\sigma^* = (U_1, U_2, \ldots, U_{n+1}, V^*)$ be a valid aggregate signature on the message set $\{m_1, m_2, \ldots, m_{n+1}\}$ for corresponding identities $\{ID_1, ID_2, \ldots, ID_{n+1}\}$ with public key $\{upk_{ID_1}, upk_{ID_2}, \ldots, upk_{ID_{n+1}}\}$ by verification. Validation of aggregate signature can be verified by the following signature.

\[
e(S^*, P) = e\left(\sum_{i=1}^{n+1} psk_{ID_i} + \sum_{i=1}^{n+1} t_i \cdot r_i \cdot P_{pub} + \sum_{i=1}^{n+1} h_i \cdot v \cdot Q + \sum_{i=1}^{n+1} k_i \cdot r_i \cdot Q, P\right)
\]
\[
= e\left(\sum_{i=1}^{n+1} (sQ_{ID_i}, P)\right) \cdot e\left(\sum_{i=1}^{n+1} t_i r_i P, sP\right) \cdot e\left(\sum_{i=1}^{n+1} h_i \cdot v \cdot P, Q\right)
\]
\[
= e\left(\sum_{i=1}^{n+1} Q_{ID_i}, P_{pub}\right) \cdot e\left(\sum_{i=1}^{n+1} t_i U_i, P_{pub}\right) \cdot e\left(\sum_{i=1}^{n+1} h_i \cdot upk_{ID_i}, Q\right)
\]
\[
= e\left(\sum_{i=1}^{n+1} Q_{ID_i} + t_i U_i, P_{pub}\right) \cdot e\left(\sum_{i=1}^{n+1} h_i \cdot upk_{ID_i} + k_i \cdot U_i, Q\right)
\]

4. Conclusion

Deng et al. designed an efficient scheme certificateless aggregate signature and prove in a random oracle model environment that no one adversary presents in the network that could forge the security of the scheme. In this paper, we describe the collision insider attack and demonstrate that Deng et al. proposed CL-AS scheme is not secure against the collision insider attack.

References


PANKAJ KUMAR: SCHOOL OF COMPUTING SCIENCE AND ENGINEERING, GALGOTIAS UNIVERSITY, NOIDA, UTTAR PRADESH, INDIA
E-mail address: pkumar240183@gmail.com

VISHNU SHARMA: SCHOOL OF COMPUTING SCIENCE AND ENGINEERING, GALGOTIAS UNIVERSITY, NOIDA, UTTAR PRADESH, INDIA
E-mail address: vishnusharma97@gmail.com

VINOD KUMAR: DEPARTMENT OF MATHEMATICS, P.G.D.A.V. COLLEGE, UNIVERSITY OF DELHI, NEW DELHI 110065, INDIA
E-mail address: vinod.iitkgp13@gmail.com