Efficient Certificateless Signcryption

Diego Aranha, Rafael Castro, Julio López, Ricardo Dahab

1 Institute of Computing – University of Campinas (UNICAMP)
CEP 13084-971 – Campinas – SP - Brazil

{dfaranha,jlopez,rdahab}@ic.unicamp.br,rafael.castro@gmail.com

1. Introduction

The conventional public key cryptography model includes a central authority that issues certificates and manages a public key infrastructure, requiring significant processing and storage capabilities. Identity-based cryptography (ID-PKC) replaces the traditional public keys with identifiers derived from users’ identities. This facilitates public key validation but introduces the key escrow of private keys by the central authority as a side-effect.

Certificateless cryptography (CL-PKC) is a novel paradigm where the generated costs are reduced without introducing key escrow of private keys.

A signcryption scheme is a technique that provides confidentiality, authentication and non-repudiation in a single integrated operation. The first concrete CL-PKC signcryption scheme was proposed recently in [Barbosa and Farshim 2008]. We propose an efficient CL-PKC signcryption scheme that supports publicly verifiable signatures, and that is more efficient than the first protocol.

2. Bilinear Pairings

Let $G_1$ and $G_2$ be additive groups of order $q$ and $G_T$ be a multiplicative group of order $q$. Let $P$ and $Q$ be the generators of $G_1$ and $G_2$ respectively. An efficiently-computable map $e : G_1 \times G_2 \rightarrow G_T$ is an admissible bilinear map if the following properties are satisfied:

1. Bilinearity: given $(Q, W) \in G_1 \times G_2$ and $(a, b) \in \mathbb{Z}_q^*$, we have:
   $$e(aQ, bW) = e(Q, W)^{ab} = e(abQ, W) = e(Q, abW).$$
2. Non-degeneracy: $e(P, Q) \neq 1_{G_T}$, where $1_{G_T}$ is the identity of the group $G_T$.

3. Efficient Signcryption

The proposed signcryption scheme is an extension of an efficient ID-PKC signcryption scheme proposed in [McCullagh and Barreto 2004], inheriting the public verification feature. Our protocol has the following algorithms:

Setup. Given a security parameter $k$, the central authority (Key Generation Center – KGC) generates a $k$-bit prime number $q$, bilinear groups $(G_1, G_2, G_T)$ of order $q$ with generators $P \in G_1$ and $Q \in G_2$, and an admissible bilinear map $e$. The KGC also chooses hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : G_T \rightarrow \{0, 1\}^n$ and $H_3 : \{0, 1\}^n \times G_1 \times G_1 \rightarrow \mathbb{Z}_q^*$, selects at random the master key $s \in \mathbb{Z}_q^*$ and computes $P_{pub} = sP$ and $g = e(P, Q)$. The KGC publishes the system parameters $(q, G_1, G_2, G_T, P, Q, e, g, P_{pub}, H_1, H_2, H_3)$ and keeps $s$ in secret.

Extract. Let $y_E$ denote $H_1(ID_E)$. Given identity $ID_A$, the KGC computes and issues to user $A$ the partial private key $D_A = (y_A + s)^{-1}Q \in G_2$;
**Keygen.** User \( A \) selects at random \( x_A \in \mathbb{Z}_q^* \) as a secret value and computes the private key \( S_A = x_A^{-1}D_A \in \mathbb{G}_2 \) and the public key \( P_A = x_A(y_AP + P_{pub}) \in \mathbb{G}_1 \). The resulting key pair is \((P_A, S_A)\). Observe that \( e(P_A, S_A) = g \).

**Signcrypt.** To signcrypt the message \( M \), user \( A \) computes:
1. \( r \leftarrow R \mathbb{Z}_q^*, u \leftarrow r^{-1}, U \leftarrow g^u; \)
2. \( C \leftarrow M \oplus H_2(U); \)
3. \( h \leftarrow H_3(C, rP_A, uP_B); \)
4. \( T \leftarrow (r + h)^{-1}S_A; \)
5. Return \((C, rP_A, uP_B, T)\).

**Unsigncrypt.** Upon reception of the signcrypted message \((C, R, S, T)\), user \( B \) computes:
1. \( h' \leftarrow H_3(C, R, S); \)
2. \( V \leftarrow e(R + h'P_A, T); \)
3. \( r' \leftarrow e(S, S_B); \)
4. \( M' \leftarrow C \oplus H_2(r'); \)
5. If \( V = g \), return \( M' \). Otherwise, return \( \perp \) indicating error.

The scheme is publicly verifiable, as the computation of \( V \) does not depend on private information. If \((C, R, S, T)\) is correct, we can see that the protocol works:
- \( V = e(R + hP_A, T) = e((r + h)P_A, (r + h)^{-1}S_A) = e(P_A, S_A) = g. \)
- \( e(S, S_B) = e(uP_B, x_B^{-1}D_B) = e(ux_B(y_BP + P_{pub}), x_B^{-1}(y_B + s)^{-1}Q) = g^u = U. \)

The computational costs of the proposed protocol and the scheme from [Barbosa and Farshim 2008] are presented in Table 1. The cost is measured in terms of bilinear pairings (\( e \)), exponentiations in \( \mathbb{G}_T(a^x) \), scalar multiplications in \( \mathbb{G}_1 \) or \( \mathbb{G}_2 \) (\( kP \)), inversions in \( \mathbb{Z}_q^* \) (\( a^{-1} \)) and hash functions (\( H \)) computations.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Protocol</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preprocessing</td>
<td>[Barbosa and Farshim 2008]</td>
<td>Proposed</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0</td>
</tr>
<tr>
<td>Signcrypt</td>
<td>[Barbosa and Farshim 2008]</td>
<td>Proposed</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0</td>
</tr>
<tr>
<td>Unsigncrypt</td>
<td>[Barbosa and Farshim 2008]</td>
<td>Proposed</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>2</td>
</tr>
</tbody>
</table>

† Two of the scalar multiplications can be simultaneous

4. **Future work**

Future works will be centered on proving the scheme security in a formal setting.

**References**
