# Efficient Certificateless Signcryption

**Diego Aranha, Rafael Castro, Julio López, Ricardo Dahab**[1]

[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 and secure CL-PKC signcryption scheme was proposed recently in [1]. We propose an efficient CL-PKC signcryption scheme that supports publicly verifiable signatures and show that it is more efficient than the first protocol. This paper fixes a previous protocol proposed in [2] and cryptanalysed in [3].

## 2. Bilinear Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additive groups of prime order $q$ and $\mathbb{G}_T$ be a multiplicative group of order $q$. Let $P$ and $Q$ be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. An efficiently-computable map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an *admissible bilinear map* if the following properties are satisfied:

1. *Bilinearity:* given $(V, W) \in \mathbb{G}_1 \times \mathbb{G}_2$ and $(a, b) \in \mathbb{Z}_q^*$, we have: $e(aV, bW) = e(V, W)^{ab} = e(abV, W) = e(V, abW)$.
2. *Non-degeneracy:* $e(P, Q) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the identity of the group $\mathbb{G}_T$.

Different pairing instantiations lead to bilinear maps with distinct performance and functionality features [4]. When $\mathbb{G}_1 = \mathbb{G}_2$ (symmetric pairing), the pairing is called a Type 1 pairing and is implemented using supersingular curves. The case $\mathbb{G}_1 \neq \mathbb{G}_2$ (asymmetric pairing) can be separated into two sub-classes: Type 2 pairings where an efficient homomorphism $\varphi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is available and Type 3 where such maps are not possible. Asymetric pairings can support a higher embedding degree tending to be faster at high security levels [5, 6].

## 3. Efficient Signcryption

The proposed signcryption scheme is an extension of an efficient ID-PKC signcryption scheme proposed in [7], inheriting the public verification feature. The scheme is also not restricted to symmetric or asymmetric pairing settings. 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 ($\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$) of order $q$ with generators $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, and an admissible bilinear map $e$. The KGC also chooses hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : \mathbb{G}_T \to \{0,1\}^n$ and $H_3 : \{0,1\}^n \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^*$, 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 $\langle q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P, Q, e, g, P_{pub}, H_1, H_2, H_3 \rangle$ and keeps $s$ in secret.

**Extract.** Let $y_E$ denote $H_1(\mathsf{ID}_E)$. Given identity $\mathsf{ID}_A$, the KGC computes and issues to user $A$ the partial private key $D_A = (y_A + s)^{-1}Q \in \mathbb{G}_2$;

**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 $P_A = x_A(y_A P + P_{pub}) \in \mathbb{G}_1$. 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. $c \leftarrow m \oplus H_2(U, P_A, \mathsf{ID}_A)$;
3. $h \leftarrow H_3(c, rP_A, uP_B, P_A, \mathsf{ID}_A)$;
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. $h' \leftarrow H_3(c, R, S, P_A, \mathsf{ID}_A)$;
2. $V \leftarrow e(R + h'P_A, T)$;
3. $r' \leftarrow e(S, S_B)$;
4. $m' \leftarrow c \oplus H_2(r', P_A, \mathsf{ID}_A)$;
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:

- $V = e(R + hP_A, T) = e((r + h)P_A, (r + h)^{-1} S_A) = e(P_A, S_A) = g$.
- $e(S, S_B) = e(uP_B, x_B^{-1} D_B) = e(ux_B(y_B P + P_{pub}), x_B^{-1}(y_B + s)^{-1} Q) = g^u = U$.

## 4. Security

In this updated version of the protocol, hash function $H_2, H_3$ were changed so both also depends on $P_A$ and $\mathsf{ID}_A$. This modification fixes both attacks presented in [3]. We present below the attacks on the previous version of the protocol and discuss why the new version is not affected.

**Attack 1:** the first attack requires a Type-I adversary who is capable of replacing public keys but does not have access to the master key $s$. The attacker forges a valid signcryption on a message $m$ from a legitimate user $A$ to another user $B$ by performing the following steps:

1. Choose randomly $r \in_R \mathbb{Z}_q^*$ and compute $u = r^{-1}$;
2. Compute $U = g^r$ and set $C = m \oplus H_2(U)$;
3. Set $T = r^{-1}Q$, $R = rP - P$ and $S = uP_B$;
4. Compute $h = H_3(c, R, S)$;
5. Replace the public key of $A$ with $P_A = h^{-1}P$.

The forged signcryption on message $m$ is $\sigma = (c, R, S, T)$. It is straightforward to see that the updated function $H_3$ avoids this attack. If $H_3$ depends on $P_A$, it is not possible to replace the public key of $A$ after the signcryption and keep the signature component valid. The hash computed by $B$ with the wrong public key will allow $B$ to reject the message.

**Attack 2:** the second attack is executed by a Type-I or Type-II adversary on the unforgeability and confidentiality of the scheme. Let $\sigma^* = (c^*, R^*, S^*, T^*)$ be the challenge signcryption on message $m_b, b \in \{0, 1\}$ sent by user $A$ to user $B$. The adversary generates a new signcryption $\sigma' = (c', R', S', T')$ on $m_b$ from an user $C$ (whose private key is known to the adversary) to the same receiver $B$ by performing the following steps:

1. Set $c' = c^*$;
2. Choose randomly $r' \in_R \mathbb{Z}_q^*$ and compute $R' = r'P_C$;
3. Set $S' = S^*$;
4. Compute $h' = H_3(c', R', S')$;
5. Set $T' = (r' + h')^{-1} S_C$.

Now the attacker can query the unsigncryption oracle for the unsigncryption of $\sigma'$. The oracle will give back the message $m_b$ because $\sigma'$ is a valid signcryption from $C$ to $B$ on $m_b$ and $\sigma' \neq \sigma$. If $H_2$ depends on $P_A$ and $\mathsf{ID}_A$, the component $c^*$ is tied to the identity and public key of the original sender $A$ and the attacker is not able to produce a new valid signcryption with $C$ as the sender. The hash computed by the receiver $B$ or the unsigncryption oracle with the wrong public key or identity will turn $\sigma'$ into the signcryption on a random message from $C$ to $B$, not providing any useful knowledge to the adversary.

## 5. Efficiency

The computational costs of the proposed protocol and the scheme from [1] 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 1. Computational cost of the protocols in operations.**

| | | Operations | | | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **Protocol** | $e$ | $kP$ | $a^x$ | $a^{-1}$ | $H$ |
| Preprocessing | [1] | 1 | 0 | 0 | 0 | 0 |
| | Proposed | 0 | 0 | 0 | 0 | 0 |
| Signcrypt | [1] | 0 | 3 + $\sigma^\dagger$ | 1 | 0 | 3 |
| | Proposed | 0 | 3 | 1 | 2 | 2 |
| Unsigncrypt | [1] | 4 | 1 | 0 | 0 | 3 |
| | Proposed | 2 | 1 | 0 | 0 | 2 |

$^\dagger$ Two of the scalar multiplications can be simultaneous

## 6. Future work

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

## References


[1] M. Barbosa P. and Farshim. Certificateless signcryption. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS 2008)*, pages 369–372, New York, NY, USA, 2008. ACM.

[2] D. F. Aranha, R. Castro, J. López, and R. Dahab. Efficient Certificateless Signcryption. In *VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG 2008)*, pages 257–258, 2008.

[3] S. Sharmila Deva Selvi, S. Sree Vivek, and C. Pandu Ragan. On the Security of Certificateless Signcryption Schemes. Cryptology ePrint Archive, Report 2009/298, 2009. `http://eprint.iacr.org/`.

[4] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for Cryptographers. *Discrete Applied Mathematics*, 156(16), 2008.

[5] D. Hankerson, A. Menezes, and M. Scott. Software Implementation of Pairings. In *Identity-Based Cryptography*, chapter 12, pages 188–206. IOS Press, 2008.

[6] D. F. Aranha, J. López, and D. Hankerson. High-Speed Parallel Software Implementation of the $\eta_T$ Pairing. In J. Pieprzyk, editor, *Cryptographers' Track at RSA Conference (CT-RSA 2010)*, volume 5985 of *LNCS*, pages 89–105. Springer, 2010.

[7] N. McCullagh and P. S. L. M. Barreto. Efficient and Forward-Secure Identity-Based Signcryption. Cryptology ePrint Archive, Report 2004/117, 2004. `http://eprint.iacr.org/`.