Identity-Based Signcryption

John Malone-Lee

University of Bristol, Department of Computer Science,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB, UK
malone@cs.bris.ac.uk

Abstract. A signcryption scheme is a scheme that provides private and authenticated delivery of messages between two parties. It does this in a more efficient manner than a straightforward composition of an encryption scheme with a signature scheme. An identity-based cryptosystem is one in which the public key may be any string (or may be derived from any string).
In this paper we propose an identity-based signcryption scheme. We give a security model for such a scheme and sketch the details of how our scheme may be proved secure in this model.

1 Introduction

Two of the most important services offered by cryptography are those of providing private and authenticated communications. Much research has been done into creating encryption schemes to meet highly developed notions of privacy [4–6, 10–13]. Similarly, designing unforgeable signature schemes to give authenticity and non-repudiation is also a well studied problem [8, 14, 17–20]. It is possible to combine encryption schemes and signature schemes, using methods such as those described in [2], to obtain private and authenticated communications.

In 1997 Zheng proposed a primitive that he called *signcryption* [23]. The idea of a signcryption scheme is to combine the functionality of an encryption scheme with that of a signature scheme. It must provide privacy; signcryptions must be unforgeable; and there must be a method to settle repudiation disputes. This must be done in a more efficient manner than a composition of an encryption scheme with a signature scheme.

In 1984 Shamir proposed the idea of *identity-based cryptography* [21]. The idea of an identity based cryptosystem is that the public key can be an arbitrary string. For such a system to work there is a *trusted authority* (TA henceforth) that generates private keys using some master key related to the global parameters for the system. The first practical identity-based cryptosystem was that of Boneh and Franklin in 2001 [7]. Since then the ideas of [7] have been used to design several other identity-based schemes [9, 15, 16, 22].

In this paper we describe an *identity-based signcryption scheme*. We give a model of security for such schemes and sketch the details of how our scheme may be proved secure in this model. We make a comparison between the efficiency of our scheme and a composition of an encryption scheme with a signature scheme.

2 Identity-Based Signcryption

An identity based signcryption scheme uses four algorithms: `Setup`, `Extract`, `Signcrypt` and `Unsigncrypt`. The functions of these algorithms are described below.

- Given a security parameter k, `Setup` is used by the TA to generates the global systems parameters `parameters`. Among the parameters produced by `Setup` is a key Q_{TA} that is made public. There is also corresponding master key t that is kept secret.
- Given a string ID representing the identity of some party, the TA uses `Extract` to generate a corresponding secret key S_{ID} which it gives to ID (here and henceforth we make no distinction between a party and its identity).
- If ID_a wishes to send a message m to ID_b it generates the appropriate ciphertext using `Signcrypt`. In this situation `Signcrypt` takes as input S_{ID_a}, ID_b and m to produce a ciphertext σ. We will assume throughout that the message space is $\{0,1\}^n$ for some $n \in \mathbb{N}$.
- If ID_b has received a ciphertext σ from ID_a it uses `Unsigncrypt` to recover the corresponding plaintext. In this situation `Unsigncrypt` takes as input ID_a, S_{ID_b} and σ to return a message m or the symbol $\perp$. The symbol $\perp$ indicates that the ciphertext was invalid.

We make the consistency constraint that if $\sigma \leftarrow \texttt{Signcrypt}(S_{ID_a}, ID_b, m)$, then $m \leftarrow \texttt{Unsigncrypt}(ID_a, S_{ID_b}, \sigma)$.

3 Security of Identity-Based Signcryption

3.1 Confidentiality

The *de facto* definition of security for public key encryption schemes is *indistinguishability of encryptions under chosen ciphertext attack* [1, 3, 5, 10, 11]. Our definition of security will be a natural adaptation of this to the identity-based setting for signcryption schemes. We will call the new form of security *indistinguishability of identity-based signcryptions under chosen ciphertext attack* (IND-ISC-CCA).

Definition 1. *IND-ISC-CCA Security*
Consider the following game played by a challenger, `C`, and an adversary, `A`.

Setup: `C` takes a security parameter k and runs `Setup` to obtain `parameters` and a master key t. It gives `parameters` to `A` and keeps t secret.

Phase 1: During this phase `A` may make the queries described below to `C`.

- Extraction queries of the form ID_i. On receiving such a query `C` runs `Extract` (ID_i) and responds with S_{ID_i}. Up to q_{id} extraction queries may be made by `A` during Phase 1 and Phase 2.
- Signcryption queries of the form (ID_i, ID_j, m). On receiving such a query `C` runs `Extract`(ID_i) followed by `Signcrypt`(S_{ID_i}, ID_j, m). The response is the resulting ciphertext. Up to q_{sc} signcryption queries may be made by `A` during Phase 1 and Phase 2.
- Unsigncryption queries of the form (ID_i, ID_j, σ). On receiving such a query `C` runs `Extract`(ID_j) followed by `Unsigncrypt`(ID_i, S_{ID_j}, σ). The response is the resulting plaintext m or the symbol $\perp$. Up to q_{usc} unsigncryption queries may be made by `A` during Phase 1 and Phase 2.

These queries may be made adaptively i.e. each query may depend on the responses to previous queries. At the end of Phase 1 `C` outputs two messages, m_0 and m_1, and two identities, ID_a and ID_b, on which it wishes to be challenged. The identities ID_a and ID_b must not have appeared in extraction queries during Phase 1.

Challenge: `C` picks a random b from $\{0, 1\}$ and runs `Extract`(ID_a) followed by `Signcrypt`(S_{ID_a}, ID_b, m_b). It returns the resulting ciphertext σ^* to `A`.

Phase 2: During this phase `A` may make more queries of the types described in Phase 1 with the restrictions below.

- The extraction queries ID_a and ID_b are not permitted.
- The unsigncryption query (ID_a, ID_b, σ^*) is not permitted.

As in Phase 1, these queries may be made adaptively.

Guess: Finally, `A` outputs a bit $b' \in \{0, 1\}$. It wins if $b' = b$.

We define the advantage of `A` to be $\mathbf{Adv}(\mathtt{A}) = |\Pr[b' = b] - \frac{1}{2}|$. We say that an identity-based signcryption scheme is IND-ISC-CCA secure if no polynomially bounded adversary has non-negligible advantage in the game described above.

□

3.2 Unforgeability

The accepted definition of security for signature schemes is *existential unforgeability under adaptive chosen message attack* [8, 14, 17, 18]. Our definition of unforgeability will be a natural adaptation of this to the identity-based setting for signcryption. We call our new definition *existential unforgeability of identity based signcryptions under adaptive chosen message attack* (EF-ISC-ACMA).

Definition 2. *EF-ISC-ACMA Security*
Consider the following game played by a challenger `C` and an adversary `A`.

Setup: `C` takes a security parameter k and runs `Setup` to obtain `parameters` and a master key t. It gives `parameters` to `A` and keeps t secret.

Attack: During this phase `A` may make the queries described below to `C`.

- Extraction queries of the form ID_i. On receiving such a query `C` runs `Extract` (ID_i) and responds with S_{ID_i}. Up to q_{id} extraction queries may be made by `A`.
- Signcryption queries of the form (ID_i, ID_j, m). On receiving such a query `C` runs `Extract`(ID_i) followed by `Signcrypt`(S_{ID_i}, ID_j, m). The response is the resulting ciphertext. Up to q_{sc} signcryption queries may be made by `A`.
- Unsigncryption queries of the form (ID_i, ID_j, σ). On receiving such a query `C` runs `Extract`(ID_j) followed by `Unsigncrypt`(ID_i, S_{ID_j}, σ). The response is the resulting plaintext m or the symbol $\perp$. Up to q_{usc} unsigncryption queries may be made by `A`.

These queries may be made adaptively i.e. each query may depend on the responses to previous queries.

Forge: Finally `A` outputs (σ^*, ID_a, ID_b), where ID_a was not an extract query during Attack. It wins if `Unsigncrypt`$(ID_a, S_{ID_b}, \sigma^*)$ does not return $\perp$.

We define the advantage of `A` to be $\mathbf{Adv}(\mathtt{A}) = \Pr[\mathtt{A} \text{ wins}]$. We say that an identity-based signcryption scheme is EF-ISC-CMA secure if no polynomially bounded adversary has non-negligible advantage in the game described above.

□

Notice in Definition 2 that we allow `A` to have seen S_{ID_b} for the identity ID_b with respect to which its forgery σ^* is valid. This is a necessary condition for an identity-based signcryption scheme to offer non-repudiation. If a receiver of signcrypted messages, ID_b in this case, is able to forge signcryptions to itself from some sender, ID_a in this case, then a third party will never be able to determine which of the two parties signcrypted a particular message.

4 An Identity-Based Signcryption Scheme

Let $(G, +)$ and $(V, \cdot)$ denote cyclic groups of prime order q. Let P be a generator of G and let $\hat{e} : G \times G \to V$ be a pairing satisfying the conditions below.

1. Bilinear: For all $P, Q \in G$ and all $a, b \in \mathbb{Z}$ we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
2. Non-degenerate: There exists $P_1, P_2 \in G$ such that $\hat{e}(P_1, P_2) \neq 1$. This means that if P is a generator of G, then $\hat{e}(P, P)$ is a generator of V.

Such groups may be realized using supersingular elliptic curves and the Weil pairing. For further details see [7, 15].

The scheme also requires three hash functions:

$$H_1 : \{0,1\}^* \to G^*, \; H_2 : \{0,1\}^* \to \mathbb{Z}_q^*, \; H_3 : \mathbb{Z}_q^* \to \{0,1\}^n.$$

Where $n \in \mathbb{Z}$ is the length of the messages to be signcrypted and G^* denotes $G \setminus \{0\}$.

Using these groups and hash functions we are now ready to describe our identity-based signcryption scheme IDSC. For a set S we use $x \stackrel{r}{\leftarrow} S$ to denote the procedure of selecting an element of S at random and assigning the value of x to this element. We use $x \leftarrow y$ to denote the procedure of assigning the value of x to the value of y. Concatenation is denoted $||$.

Scheme 1 *IDSC*
Once `Setup` has been run we will assume that `parameters` are available to all algorithms and so it is not necessary to provide them as input. We will assume that $P, \hat{e}, H_1, H_2$ and H_3 are as above and that they are all produced by `Setup` to form part of `parameters`, although we do not include this in the description below.

`Setup`	`Extract`(ID)
$t \stackrel{r}{\leftarrow} \mathbb{Z}_q^*$	$Q_{ID} \leftarrow H_1(ID)$
$Q_{TA} \leftarrow tP$	$S_{ID} \leftarrow tQ_{ID}$
`Signcrypt`(S_{ID_a}, ID_b, m)	`Unsigncrypt`(ID_a, S_{ID_b}, σ)
$Q_{ID_b} \leftarrow H_1(ID_b)$	$Q_{ID_a} \leftarrow H_1(ID_a)$
$x \stackrel{r}{\leftarrow} \mathbb{Z}_q^*$	Parse σ as (c, U, V)
$U \leftarrow xP$	$y \leftarrow \hat{e}(S_{ID_b}, U)$
$r \leftarrow H_2(U\|\|m)$	$\kappa \leftarrow y$
$W \leftarrow xQ_{TA}$	$m \leftarrow \kappa \oplus c$
$V \leftarrow rS_{ID_a} + W$	$r \leftarrow H_2(U\|\|m)$
$y \leftarrow \hat{e}(W, Q_{ID_b})$	If $\hat{e}(V, P) \neq \hat{e}(Q_{ID_a}, Q_{TA})^r \cdot \hat{e}(U, Q_{TA})$
$\kappa \leftarrow H_3(y)$	return $\perp$
$c \leftarrow \kappa \oplus m$	Return m
$\sigma \leftarrow (c, U, V)$	

□

Note that IDSC offers non-repudiation. To see why suppose that ID_b receives m from ID_a signcrypted as (c, U, V). For this to be independently verified ID_b surrenders m to a third party who checks that $\hat{e}(V, P) = \hat{e}(Q_{ID_a}, Q_{TA})^r \cdot \hat{e}(U, Q_{TA})$ where $r \leftarrow H_2(U||m)$.

5 Analysis of IDSC

5.1 Efficiency

We compare the efficiency of IDSC with that of the encrypt-then-sign method of [2] using the encryption scheme of [7], denoted BF, and the signature scheme of [9], denoted CC.

Let us first consider the size of the cryptograms of the two methods when a message of n bits is to be sent. In the table below $|G|$ denotes the bit length of an element of G.

scheme	size of cryptogram
IDSC	$n + 2\|G\|$
BF and CC	$2n + 3\|G\|$

As we can see from the table above, IDSC signcryptions are considerably more compact than those produced using BF and CC.

Let us now consider the computation required by the two methods. In the table below we enumerate the various operations necessary for each.

	IDSC		CC and BF	
operation type	signcrypt	unsigncrypt	encrypt/sign	verify/decrypt
$\hat{e}$ evaluation	1	4	1	3
multiplication in G	3	0	3	2
exponentiation in V	0	1	1	0

If we regard the operations above as the expensive ones in the generation and validation of ciphertexts we see that we have saved one operation in the generation of ciphertexts and we have the same number in validation.

5.2 Security

It is straightforward to give a proof of the IND-ISC-CCA security of IBSC in the random oracle model [4] under the *bilinear Diffie-Hellman assumption* [7]. Roughly speaking this is the assumption is that given P, aP, bP and cP it is not possible to compute $\hat{e}(P,P)^{abc}$. To construct such a proof one sets $U^* \leftarrow aP$ (where the challenge ciphertext σ^* is (c^*, U^*, V^*)), $Q_{ID_b} \leftarrow bP$ and $Q_{TA} \leftarrow cP$. Now, if H_3 is a random oracle, an adversary $\mathtt{A}$ can have no advantage in distinguishing encryptions unless it makes the H_3 query $\hat{e}(P,P)^{abc}$.

To give a proof of the EF-ISC-ACMA security of IBSC in the random oracle model it is possible to use the method of [15] which is based on [18]. This proof is uses the assumption that given $P, Q \in G$ and $R = sP$, it is not possible to compute sQ. Note that if this were possible then one could compute the private key S_{ID_x} of ID_x without going to the TA.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In *Topics in Cryptology - CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158. Springer-Verlag, 2001.
2. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998.

4. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
5. M. Bellare and P. Rogaway. Optimal asymmetric encryption - how to encrypt with RSA. In *Advances in Cryptology - EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1994.
6. D. Boneh. Simplified OAEP for the RSA and Rabin functions. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 275–291. Springer-Verlag, 2001.
7. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001. Full version available at http://eprint.iacr.org/2001/090/.
8. E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC'2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 276–292. Springer-Verlag, 2000.
9. J. C. Cha and J. H. Cheon. An identity-based signature from gap Diffie-Hellman groups. Available at http://eprint.iacr.org/2002/018/.
10. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer-Verlag, 1998.
11. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. Available at http://eprint.iacr.org/search.pl, 2001.
12. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer-Verlag, 2002.
13. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
14. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *S. I. A. M. Journal on Computing*, 17(2):281–308, 1988.
15. F. Hess. Efficient identity based signature schemes based on pairings. To appear SAC 2002.
16. K. G. Patterson. ID-based signatures from pairings on elliptic curves. Available at http://eprint.iacr.org/2002/004/.
17. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.
18. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
19. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 387–394. ACM Press, 1990.
20. C. P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 235–251. Springer-Verlag, 1990.

21. A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 0193 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
22. N. P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. *Electronic Letters*, 38(13):630–632, 2002.
23. Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.