Hybrid Signcryption Schemes With Outsider Security

(Extended Abstract)

Alexander W. Dent

Information Security Group,
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, U.K.
a.dent@rhul.ac.uk

Abstract. This paper expands the notion of a KEM–DEM hybrid encryption scheme to the signcryption setting by introducing the notion of a signcryption KEM, a signcryption DEM and a hybrid signcryption scheme. We present the security criteria that a signcryption KEM and DEM must satisfy in order that the overall signcryption scheme is secure against outsider attacks. We also present ECISS–KEM — a simple, efficient and provably secure example of a signcryption KEM. Lastly, we briefly discuss the problems associated with using KEMs in key establishment protocols.

1 Introduction

Hybrid cryptography as the branch of asymmetric cryptography that makes use of keyed symmetric cryptosystems as black-box algorithms with certain security properties. The critical point of this definition is that it is the properties of the symmetric cryptosystem that are used to construct the asymmetric scheme, rather than the technical details about the way in which the symmetric algorithm achieves these security properties.

Traditionally, hybrid cryptography has been concerned with building asymmetric encryption schemes; for example, the ECIES scheme [1]. Typically, in these cases, a symmetric encryption scheme (such as a block cipher in a particular mode of operation) has been used as part of an asymmetric encryption scheme in order to overcome the problems associated with encrypting long messages with 'pure' asymmetric techniques. More recently, symmetric encryption schemes have been used to similar effect in signcryption schemes [2, 10].

Another recent advance in hybrid cryptography is the development of the KEM–DEM model for hybrid encryption algorithms [8, 16]. This model splits a hybrid encryption scheme into two distinct components: an asymmetric key encapsulation mechanism (KEM) and a symmetric data encapsulation mechanisms (DEM). Whilst the KEM–DEM model does not model all possible hybrid encryption schemes, and there are several examples of hybrid encryption schemes that do not fit into the KEM–DEM model, it does have the advantage that

it allows the security requirements of the asymmetric and symmetric parts of the scheme to be completely separated and studied independently. This model demonstrates what should be an overriding principle of hybrid cryptography: it is not necessary for an asymmetric scheme to fully involve itself in the details of providing a security service — the security service can be provided by a symmetric scheme provided the asymmetric scheme is in full control of that process (say, by generating the secret key that the symmetric scheme uses).

In this paper we will apply this separation principle to signcryption schemes that have outsider security. A signcryption scheme is outsider secure if it is secure against attacks made by any third party (i.e. attacks made by an entity who is neither the sender nor the receiver) [3]. This is a weaker notion of security than has been traditionally dealt with by signcryption schemes, a notion known as insider security. Signcryption scheme with outsider security do not not provide any kind of non-repudiation guarantee[1], but, as is argued in [3], this is not required for most applications[2]. As we shall note in Section 8, the standard KEM/DEM construction cannot be used to produce a signcryption scheme with insider security. Hybrid signcryption schemes with insider security are considered in a companion paper [9].

As in the encryption setting, we will provide a generic model for a hybrid signcryption scheme that fully separates the asymmetric and symmetric parts of the scheme, and define security criteria that each parts should meet if the overall signcryption scheme is to be secure. We will also propose a concrete example of a "signcryption KEM" (the asymmetric part of the generic hybrid signcryption scheme) and prove its security in the random oracle model. Lastly, we will discuss a question that has been asked several times since the proposal of the KEM–DEM model: is it possible to use an encryption KEM as a key establishment mechanism?

2 Signcryption Schemes with Outsider Security

A signcryption scheme [17] is an asymmetric scheme that combines the advantages of an asymmetric encryption scheme with most of those of a digital signature scheme, i.e. the scheme transmits messages confidentially and in a manner

[1] Of course, most signcryption schemes do not offer non-repudiation to a third party [14], but a signcryption scheme that is only secure against outsider attacks can *never* offer a non-repudiation service.

[2] It can be argued that hybrid signcryption schemes with outsider security serve no purpose as the same effect can be achieved using authenticated key agreement and symmetric authenticated encryption techniques. This argument similarly applies to hybrid encryption, and, in the author's opinion, somewhat misses the point. Hybrid encryption and signcryption allows us to decouple the maximum message size from the security level that the asymmetric scheme affords. In most 'pure' asymmetric algorithms, a long message can only be sent using large values for the public key, thus resulting in high computational costs and an unnecessarily high security level. Just as hybrid encryption schemes have been found to be useful in the real world, one can expect hybrid signcryption schemes to find similar real-world uses.

in which the integrity is protected and the origin can be authenticated. It may be advantageous for a signcryption scheme to also provide a non-repudiation service; however, there are inherent problems with providing such service in this setting [14].

For our purposes a signcryption scheme will consist of five algorithms:

1. A probabilistic polynomial-time common key generation algorithm, $\mathcal{G}_c$. It takes as input a security parameter 1^k and return some global information (parameters) I.
2. A probabilistic polynomial-time sender key generation algorithm, $\mathcal{G}_s$. It takes as input the global information I and outputs a public/private key pair (pk_s, sk_s) for a party who wishes to send signcrypted messages.
3. A probabilistic polynomial-time receiver key generation algorithm, $\mathcal{G}_r$. It takes as input the global information I and outputs a public/private key pair (pk_r, sk_r) for a party who wishes to be able to receive signcrypted messages. Hence, a party who wishes to be able to both send and receive signcrypted messages will require two key-pairs: one for use when sending messages and one for use when receiving them.
4. A probabilistic polynomial-time generation-encryption algorithm, $\mathcal{E}$. It takes as input a message m from some message space $\mathcal{M}$, the private key of the sender sk_s and the public key of the receiver pk_r; and outputs a signcryption $C = \mathcal{E}(sk_s, pk_r, m)$ in some ciphertext space $\mathcal{C}$.
5. A deterministic polynomial-time verification-decryption algorithm, $\mathcal{D}$. It takes as input a signcryption $C \in \mathcal{C}$, the public key of the sender pk_s and the private key of the receiver sk_r; and outputs either a message $m = \mathcal{D}(pk_s, sk_r, C)$ or the error symbol $\perp$.

We require that any signcryption scheme is *sound*, i.e. that for almost all sender key pairs (pk_s, sk_s) and receiver key pairs (pk_r, sk_r) we have $m = \mathcal{D}(pk_s, sk_r, C)$ for almost all ciphertexts $C = \mathcal{E}(sk_s, pk_r, m)$. This definition of a signcryption scheme is essentially adapted from An [2].

We take our security notion for a signcryption scheme from An, Dodis and Rabin [3]. An *et al.* differentiate between attacks on a signcryption scheme that can be made by entities who are not involved in a particular communication (*outsiders*) and attacks that can be made by entities that are involved in a particular communication (*insiders*). A signcryption scheme that resists all attacks made by outsiders is said to be *outsider secure*.

When considering the security of a signcryption scheme we must consider its ability to resist two different classes of attacks: attacks against the confidentiality of a message and attacks against the integrity/authenticity of a message. Both of these security requirements are defined in terms of games played between a hypothetical attacker and challenger, where a signcryption scheme is secure if and only if the probability that an attacker wins the game, or the attacker's advantage in winning the game, is "negligible". Hence, we must begin by defining the term "negligible".

Definition 1. *A function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible if for every polynomial p there exists an integer N_p such that $|f(n)| \leq 1/|p(n)|$ for all $n \geq N_p$.*

Confidentiality

The notion of confidentiality for a signcryption scheme is similar to that of an asymmetric encryption scheme. The attack model is defined in terms of a game played between a hypothetical challenger and a two-stage attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. For a given security parameter k:

1. The challenger generates some global information I by running the common key generation algorithm $\mathcal{G}_c(1^k)$; a valid sender key pair (pk_s, sk_s) by running the sender key generation algorithm $\mathcal{G}_s(I)$; and a valid receiver key pair (pk_r, sk_r) by running the receiver key generation algorithm $\mathcal{G}_r(I)$.
2. The attacker runs $\mathcal{A}_1$ on the input (pk_r, pk_s). This algorithm outputs two equal length messages, m_0 and m_1, and some state information *state*. During its execution, $\mathcal{A}_1$ can query a generation-encryption oracle that will, if given a message $m \in \mathcal{M}$, return $\mathcal{E}(sk_s, pk_r, m)$, and a verification-decryption oracle that will, if given a signcryption $C \in \mathcal{C}$, return $\mathcal{D}(pk_s, sk_r, C)$.
3. The challenger picks a bit $b \in \{0, 1\}$ uniformly at random, and computes the challenge signcryption $C^* = \mathcal{E}(sk_s, pk_r, m_b)$.
4. The attacker runs $\mathcal{A}_2$ on the input $(C^*, state)$. The algorithm outputs a guess b' for b. During its execution, $\mathcal{A}_2$ can query a generation-encryption oracle and a verification-decryption oracle as above, but with the restriction that $\mathcal{A}_2$ is not allowed to query the verification-decryption oracle on the challenge ciphertext C^*.

The attacker wins the game if $b' = b$. The attacker's advantage is defined to be:

$$|Pr[b = b'] - 1/2| \,. \tag{1}$$

Definition 2 (IND-CCA security). *A signcryption scheme is said to IND-CCA secure if, for all polynomial polynomial-time attackers $\mathcal{A}$, the advantage that $\mathcal{A}$ has in winning the above game is negligible as a function of the security parameter k.*

Integrity/Authenticity

The notion of integrity for a signcryption scheme is similar to that of a digital signature scheme. The attack model is defined in terms of a game played between a hypothetical challenger and an attacker $\mathcal{A}$. For a given security parameter k:

1. The challenger generates some global information I by running the common key generation algorithm $\mathcal{G}_c(1^k)$; a valid sender key pair (pk_s, sk_s) by running the sender key generation algorithm $\mathcal{G}_s(I)$; and a valid receiver key pair (pk_r, sk_r) by running the receiver key generation algorithm $\mathcal{G}_r(I)$.
2. The attacker runs $\mathcal{A}$ on the input (pk_r, pk_s). This algorithm outputs a possible signcryption C^*. During its execution, $\mathcal{A}$ can query a generation-encryption oracle that will, if given a message $m \in \mathcal{M}$, return $\mathcal{E}(sk_s, pk_r, m)$, and a verification-decryption oracle that will, if given a signcryption $C \in \mathcal{C}$, return $\mathcal{D}(pk_s, sk_r, C)$.

The attacker wins the game if $\mathcal{D}(pk_s, sk_r, C^*) = m \neq \perp$ and $\mathcal{A}$ never received C^* as a response from generation-encryption oracle.[3]

Definition 3 (INT-CCA security). *A signcryption scheme is said to be INT-CCA secure if, for all polynomial-time attackers $\mathcal{A}$, the probability that $\mathcal{A}$ wins the above game in negligible as a function of the security parameter k.*

It is easy to see that a signcryption scheme that is both IND-CCA secure and INT-CCA secure maintains both the confidentiality and the integrity/authenticity of a message in the face of any attack by an outsider. Therefore, we define:

Definition 4 (Outsider security). *A signcryption scheme is said to be outsider secure if it is IND-CCA secure and INT-CCA secure.*

3 Hybrid Signcryption Schemes

A hybrid signcryption scheme can be formed from a "signcryption KEM" and a "signcryption DEM" in the same manner as a hybrid encryption scheme can be formed from a standard (encryption) KEM and DEM. That is to say that we may construct a hybrid signcryption scheme from an asymmetric part, that takes a private and a public key as input and outputs a suitably sized random symmetric key along with an encapsulation of the key; and a symmetric part, that takes as input a message and a symmetric key and outputs an authenticated encryption of that message.

Definition 5 (Signcryption KEM). *A signcryption KEM is a 5-tuple of polynomial-time algorithms:*

1. *A probabilistic polynomial-time common key generation algorithm, Gen_c. It takes as input a security parameter 1^k and return some global information (parameters) I.*
2. *A probabilistic polynomial-time sender key generation algorithm, Gen_s. It takes as input the global information I and outputs a public/private key pair (pk_s, sk_s) for a party who wishes to send signcrypted messages.*
3. *A probabilistic polynomial-time receiver key generation algorithm, Gen_r. It takes as input the global information I and outputs a public/private key pair (pk_r, sk_r) for a party who wishes to be able to receive signcrypted messages.*
4. *A probabilistic polynomial-time key encapsulation algorithm, Encap. It takes as input a sender's private key sk_s and a receiver's public key pk_r; and outputs a symmetric key K and an encapsulation of that key C. We denote this as $(K, C) = Encap(sk_s, pk_r)$.*

[3] This is sometimes known "strong unforgeability" in order to differentiate it from "weak unforgeability", where an attacker is only deemed to have won if $\mathcal{D}(pk_s, sk_r, C^*) = m \neq \perp$ and $\mathcal{A}$ never submitted m to the generation-encryption oracle.

5. *A deterministic polynomial-time key decapsulation algorithm, Decap. It takes as input a sender's public key pk_s, a receiver's private key sk_r and an encapsulation of a key C; and outputs either a symmetric key K or the error symbol $\perp$. We denote this as $K = Decap(pk_s, sk_r, C)$.*

We require that any signcryption KEM be sound, i.e. that for almost all sender key pairs (pk_s, sk_s) and receiver key pairs (pk_r, sk_r) we have $K = Decap(pk_s, sk_r, C)$ for almost all $(K, C) = Encap(sk_s, pk_r)$.

Definition 6 (Signcryption DEM). *A signcryption DEM is a pair of polynomial-time algorithms:*

1. *A deterministic encryption algorithm,* ENC*, which takes as input a message $m \in \{0,1\}^*$ of any length and a symmetric key K of some pre-determined length, and outputs an encryption $C = \text{ENC}_K(m)$ of that message.*
2. *A deterministic decryption algorithm,* DEC*, which takes as input a ciphertext $C \in \{0,1\}^*$ of any length and a symmetric key K of some pre-determined length, and outputs either a message $m = \text{DEC}_K(C)$ or the error symbol $\perp$.*

We require that any signcryption DEM be sound, in the sense that for every key K of the correct length, $m = \text{DEC}_K(\text{ENC}_K(m))$.

We combine a signcryption KEM and a signcryption DEM to form a hybrid signcryption scheme. As in the encryption case, we note that this is only one way in which a hybrid signcryption scheme may be formed: other hybrid signcryption schemes can be constructed that do not fit into this KEM–DEM model.

Definition 7 (KEM–DEM hybrid signcryption scheme). *Suppose that $(Gen_c, Gen_s, Gen_r, Encap, Decap)$ is a signcryption KEM,* (ENC, DEC) *is a signcryption DEM, and that, for all security parameters k, the keys produced by the signcryption KEM are of the correct length to be used by the signcryption DEM. We may then construct a signcryption scheme $(\mathcal{G}_c, \mathcal{G}_s, \mathcal{G}_r, \mathcal{E}, \mathcal{D})$ as follows:*

- *The key generation algorithms $(\mathcal{G}_c, \mathcal{G}_s, \mathcal{G}_r)$ are given by the key generation algorithms for the signcryption KEM (Gen_c, Gen_s, Gen_r).*
- *The generation-encryption algorithm $\mathcal{E}$ for a message m, a sender's private key sk_s and a receiver's public key pk_r is given by:*
  1. *Set $(K, C_1) = Encap(sk_s, pk_r)$.*
  2. *Set $C_2 = \text{ENC}_K(m)$.*
  3. *Output (C_1, C_2).*
- *The verification-decryption algorithm $\mathcal{D}$ for a signcryption (C_1, C_2), a sender's public key pk_s and a receiver's private key sk_r is given by:*
  1. *Set $K = Decap(pk_s, sk_r, C_1)$. If $K = \perp$ then output $\perp$ and stop.*
  2. *Set $m = \text{DEC}_K(C_2)$. If $m = \perp$ then output $\perp$ and stop.*
  3. *Output m.*

This construction is a sound signcryption scheme due to the soundness of the signcryption KEM and DEM.

There is only one existing signcryption scheme that can naturally be described as a KEM–DEM construction, and that is the DHETM scheme proposed by An [2]. Although, it should be noted however that the KEM part of DHETM will not meet the security criteria we propose, hence the results of this paper are not relevant to that scheme.

4 The Security Criteria for a Signcryption KEM

The advantage of a KEM–DEM construction is that it allows the security conditions of the KEM and the DEM to be assessed independently. We actually require that the KEM satisfy two security criteria: an indistinguishability criteria which is required for confidentiality and a Left-or-Right criteria that is required for integrity.

Indistinguishability

We begin by describing the security criterion that a signcryption KEM must satisfy if it is to provide a confidentiality service. This criterion is essentially the same as is required for an encryption KEM. The only difference between the two cases is that we must explicitly give the attacker access to an encapsulation oracle in the signcryption setting.

We define a signcryption KEM to indistinguishable, or IND-CCA secure, in terms of a game played between a challenger and a two-stage attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. For a given security parameter, the game runs as follows.

1. The challenger generates some global information I by running $Gen_c(1^k)$, a valid sender public/private key pair (pk_s, sk_s) by running $Gen_s(I)$, and a valid receiver public/private key pair (pk_r, sk_r) by running $Gen_r(I)$.
2. The attacker runs $\mathcal{A}_1$ on the input (pk_s, pk_r). It terminates by outputting some state information *state*. During this phase the attacker can query both an encapsulation oracle, which responds by returning $(K, C) = Encap(sk_s, pk_r)$, and a decapsulation oracle on an input C, which responds by returning $K = Decap(pk_s, sk_r, C)$.
3. The challenger generates a valid encapsulation (K_0, C^*) using $Encap(sk_s, pk_r)$. It also generates a random key K_1 of the same length as K_0. Next it chooses a bit $b \in \{0, 1\}$ uniformly at random and sets $K^* = K_b$. The challenge encapsulation is (K^*, C^*).
4. The attacker runs $\mathcal{A}_2$ on the input (K^*, C^*) and *state*. It terminates by outputting a guess b' for b. During this phase the attacker can query both an encapsulation oracle and a decapsulation oracle as above, with the exception that the decapsulation oracle cannot be queried on the challenge encapsulation C^*.

The attacker wins the game if $b = b'$. $\mathcal{A}$'s advantage is defined to be:

$$|Pr[b = b'] - 1/2| \,. \tag{2}$$

Definition 8. *A signcryption KEM is IND-CCA secure if, for every polynomial-time attacker $\mathcal{A}$, $\mathcal{A}$'s advantage in winning the above game is negligible as a function of the security parameter k.*

Left-or-Right Security

We now define what it means for a signcryption KEM to be indistinguishable from an ideal signcryption KEM. This security notion is related to the notion of Left-or-Right (LoR) security for a symmetric encryption scheme [4]. We define the ideal version of a signcryption KEM (Gen_c, Gen_s, Gen_r, $Encap$, $Decap$) to be the 5-tuple of state-based algorithms ($Sim.Gen_c$, Gen_s, Gen_r, $Sim.Encap$, $Sim.Decap$) where:

- The simulated common key generation algorithm, $Sim.Gen_c$, both runs Gen_c on the input 1^k to generate some global information I which we will be used to construct the sender and receiver public-keys, and sets up a list $KeyList$ which is initially empty.
- The simulated encapsulation algorithm, $Sim.Encap$, takes as input the pair (sk_s, pk_r) and runs as follows:
  1. Set $(K_0, C) = Encap(sk_s, pk_r)$.
  2. If there exists a pair (K_1, C) on $KeyList$ then return (K_1, C).
  3. Otherwise, generate a random symmetric key K_1 of an appropriate length, add (K_1, C) to $KeyList$ and return (K_1, C).
- The simulated decapsulation algorithm, $Sim.Decap$, takes as input the pair (pk_s, sk_r) and a signcryption C, and runs as follows:
  1. If there exists a pair (K, C) on $KeyList$ then return (K, C).
  2. If $Decap(pk_s, sk_r, C) = \perp$ then return $\perp$.
  3. Otherwise, generate a random symmetric key K of an appropriate length, add (K, C) to $KeyList$ and return K.

A signcryption KEM is said to be Left-or-Right secure if no polynomial-time attacker can distinguish between an execution where it has access to the proper signcryption KEM, and an execution where it has access to the ideal version of a signcryption KEM. We define the LoR-CCA game, for a given security parameter k, as follows:

1. The challenger picks a bit $b \in \{0, 1\}$ uniformly at random.
2. The challenger generates some global state information I either by running $Gen_c(1^k)$ if $b = 0$ or by running $Sim.Gen_c(1^k)$ if $b = 1$. The challenger also generates a valid sender public/private key pair (pk_s, sk_s) by running $Gen_s(I)$; and a valid receiver public/private key pair (pk_r, sk_r) by running $Gen_r(I)$.
3. The attacker runs $\mathcal{A}$ on the input (pk_s, pk_r). During its execution, $\mathcal{A}$ may query an encapsulation and a decapsulation oracle. If $b = 0$ then the responses to $\mathcal{A}$'s queries are computed using an encapsulation and decapsulation oracle in the normal way. If $b = 1$ then the responses to $\mathcal{A}$'s queries are computed using the ideal encapsulation and decapsulation oracles described above. $\mathcal{A}$ terminates by outputting a guess b' for b.

$\mathcal{A}$ wins the game if $b = b'$. $\mathcal{A}$'s advantage in winning the LoR-CCA2 game is given:

$$|Pr[b = b'] - 1/2| \,. \tag{3}$$

Definition 9. *A signcryption KEM is said to be Left-or-Right (LoR-CCA) secure if, for every polynomial-time attacker $\mathcal{A}$, $\mathcal{A}$'s advantage in winning the above game is negligible as a function of the security parameter k.*

It may be a little difficult to see why this security notion means that a signcryption KEM provides message integrity — intuitively, one may have expected a definition which involved an attacker trying to produce a valid symmetric key/encapsulation pair which has not been returned by the encapsulation oracle. In fact, if an attacker can do this then he can break the LoR security of the KEM by submitting such an encapsulation to the decapsulation oracle and comparing the result to the key that he expected to obtain. If the two keys are the same then the attacker can conclude that the oracles are the correct versions of the encapsulation and decapsulation algorithms, if the two keys are different then the attacker can conclude that the oracles are idealised versions of the encapsulation and decapsulation oracles. Left-or-Right security is a stronger notion of security than traditional unforgeability.

5 The Security Criteria for a Signcryption DEM

The security criteria for a signcryption DEM are a lot more intuitive than those for a signcryption KEM. Again, we must split the criteria into those required to give confidentiality and those required to give integrity/origin authentication.

Confidentiality

For confidentiality, a signcryption DEM must be IND secure in the one-time sense [8]. We define the IND security for a signcryption DEM in terms of a game played between a challenger and an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The game runs as follows:

1. The challenger randomly generates a symmetric key K of the appropriate length for the security parameter.
2. The attacker runs $\mathcal{A}_1$ on the input 1^k. This algorithm outputs two equal length messages, m_0 and m_1, as well as some state information *state*.
3. The challenger chooses a bit $b \in \{0, 1\}$ uniformly at random, and computes the challenge ciphertext $C^* = \textsc{Enc}_K(m_b)$.
4. The attacker runs $\mathcal{A}_2$ on the input $(C^*, state)$. This algorithm outputs a guess b' for b. During its execution $\mathcal{A}_2$ can query a decryption oracle that will, if queried with a ciphertext $C \neq C^*$, return $\textsc{Dec}_K(C)$.

The attacker wins the game if $b = b'$. $\mathcal{A}$'s advantage is given by:

$$|Pr[b = b'] - 1/2| \tag{4}$$

Definition 10. *A signcryption DEM is indistinguishable (IND-CCA) secure if, for every polynomial-time attacker $\mathcal{A}$, $\mathcal{A}$'s advantage in winning the above game is negligible as a function of the security parameter k.*

Integrity/Authenticity

The property of a signcryption DEM that gives it its integrity/authentication security is also a simple extension of the normal definitions. We define the INT-CCA (integrity) security for a signcryption DEM in terms of a game played between a challenger and an attacker $\mathcal{A}$. The game runs as follows:

1. The challenger randomly generates a symmetric key K of the appropriate length for the security parameter.
2. The attacker runs $\mathcal{A}$ on the input 1^k. This algorithm outputs a ciphertext C'. During its execution $\mathcal{A}$ may query a decryption oracle that will, on input of a ciphertext C, return $\textsc{Dec}_K(C)$; and an encryption oracle that will, on input of a message m, return $\textsc{Enc}_K(m)$.

The attacker wins the game if $\textsc{Dec}_K(C') \neq \perp$ and C' was never a response of the encryption oracle.

Definition 11. *A signcryption DEM is integrally secure (INT-CCA) if, for every polynomial-time attacker $\mathcal{A}$, the probability $\mathcal{A}$ wins the above game is negligible as a function of the security parameter k.*

We note that all of the usual authenticated encryption modes, including the Encrypt-then-MAC scheme discussed in Bellare and Namprempre [5] and used as a DEM by Cramer and Shoup [8], as well as the newer authentication modes such as EAX [6] and OCB [15], satisfy these security criteria.

We now state our two main results:

Theorem 1 (Confidentiality). *Suppose a hybrid signcryption scheme is composed of a signcryption KEM and a signcryption DEM. If the signcryption KEM is IND-CCA secure and the signcryption DEM is IND-CCA secure, then the hybrid signcryption scheme is IND-CCA secure (i.e. confidential).*

Theorem 2 (Integrity/Authenticity). *Suppose a hybrid signcryption scheme is composed of a signcryption KEM and a signcryption DEM. If the signcryption KEM is LoR-CCA secure and the signcryption DEM is INT-CCA secure, then the hybrid signcryption scheme is INT-CCA secure (i.e. integral and authentic).*

6 ECISS–KEM

So far we have shown that a secure signcryption scheme can be formed from suitably secure signcryption KEMs and DEMs, and that suitably secure signcryption DEMs exist. In this section we will present a secure signcryption KEM, thus demonstrating the overall feasibility of building hybrid signcryption schemes.

The scheme that we present here is similar to the ECIES-KEM scheme [12], which is based on the original DHAES scheme of Abdalla *et al.* [1]. We therefore name the scheme the Elliptic Curve Integrated Signcryption Scheme KEM (ECISS–KEM). ECISS–KEM consists of the following five algorithms:

– *Common key generation algorithm.* This algorithm takes the security parameter 1^k as input and outputs a triple (G, P, q) where G is a description of a suitably large additive cyclic group, P is a generator for that group and q is the prime order of the group.
– *Sender key generation algorithm.* This algorithm picks an integer $1 \leq s \leq q-1$ uniformly at random, sets $P_s = sP$ then outputs the public key (G, P, q, P_s) and the private key (G, P, q, s).
– *Receiver key generation algorithm.* This algorithm picks an integer $1 \leq r \leq q-1$ uniformly at random, sets $P_r = rP$ then outputs the public key (G, P, q, P_r) and the private key (G, P, q, r).
– *Signing-Encryption algorithm.* This algorithm works as follows:
  1. Choose an element $1 \leq t \leq q-1$ uniformly at random.
  2. Set $K = \mathit{Hash}(sP_r + tP)$.
  3. Set $C_1 = tP$.
  4. Output (K, C_1).
– *Verification-Decryption algorithm.* This algorithm works as follows.
  1. Set $K = \mathit{Hash}(rP_s + C_1)$.
  2. Output K.

The security of this scheme is based on the intractability of the Diffie-Hellman problem.

Definition 12. *Let G be a cyclic group with prime order q (and with the group action written additively), and let P be a generator for G. The computational Diffie–Hellman problem (CDH problem) is the problem of finding abP when given (aP, bP). We assume that a and b are chosen uniformly at random from the set $\{1, \ldots, q-1\}$.*

The proofs of the security for this algorithm are given in the full version of this paper; however, we will sketch the idea behind the security proofs. The main idea is that if we model the hash function as a random oracle then we are unable to tell the difference between the real decapsulation K of an encapsulation C and a randomly generated symmetric key unless we query the hash function (random) oracle on $sP_r + C = rP_s + C$. Therefore, if an attacker is to have any kind of advantage in either the IND-CCA or LoR-CCA games then it must make at least one such query.

However, if we set $P_s = aP$ and $P_r = bP$ for randomly generated and unknown values $1 \leq a, b \leq q-1$ then finding such a relationship between encapsulation/decapsulation oracle queries and hash oracle queries allows us to compute $sP_r = rP_s = abP$ and therefore solve a instance of the computational Diffie-Hellman problem.

Theorem 3. *The ECISS–KEM signcryption KEM is outsider secure provided the computational Diffie-Hellman problem is intractable on the group G.*

Potential problems with ECISS–KEM

We can view ECISS-KEM as producing a symmetric key by hashing a shared secret srP offset by a random value tP chosen by the sender. It is easy to see

that if an attacker recovers $srP + tP$ then they can easily recover srP and break the scheme in perpetuity. Hence, it is of the utmost importance that an implementation of the scheme keeps the value $srP + tP$ confidential.

This potential weakness could be avoided if the offset value was not easily computable by the attacker. For example, one could have an encapsulation algorithm that worked as follows:

1. Choose an element $1 \leq t \leq q - 1$ uniformly at random.
2. Set $K = Hash(sP_r + tP_r)$.
3. Set $C_1 = tP$.
4. Output (K, C_1).

This would mean that an attacker that discovered the value $sP_r + tP_r = srP + trP$ would only be able to recover the single message for which that value is used to produce the symmetric key, rather than break the scheme completely. However, precisely because it is not easy to compute srP from $sP_r + tP_r$, it becomes a lot more difficult to produce a proof of Left-or-Right security[4] for such a scheme: it is necessary to reduce the security of the scheme to a non-standard assumption. Whether an implementor wishes to use a scheme that reduces to a trusted security assumption but has a potential weakness if the security model is invalid, or use a scheme that appears more secure but reduces to an untrusted security assumption, is a very arguable issue. Some arguments about this issue have been put forward by Koblitz and Menezes [13].

7 Using KEMs as Key Establishment Mechanisms

One question that has been repeatedly asked since the inception of key encapsulation mechanisms has been "Can we use an (encryption) KEM as a key agreement mechanism?" Certainly KEMs exhibit the main property that we expect an asymmetric key agreement mechanism to have: they allow remote users to pass messages between them in such a way that both users can derive a symmetric key in a suitably secure way. The simplest form of this idea is for a sender (A) to use an encryption KEM and the public key of the receiver (B) to produce a symmetric key and an encapsulation of that key (K, C), and to send the encapsulation C to the receiver who could then recover the symmetric key by running the decapsulation algorithm using their private key. Indeed, if the KEM in question is ECIES-KEM then the resulting key agreement scheme is a standardised form of the Diffie-Hellman key agreement protocol [11].

The problem with key agreement mechanisms of this form is that they do not provide any kind of origin authentication or a guarantee of freshness, i.e. there is no way that B can know that they are involved in a key agreement protocol with A rather than some malicious entity claiming to be A, nor can they be sure

[4] An efficient proof of IND-CCA2 security that reduces the security of the scheme to the gap Diffie-Hellman assumption (a well-known variant of the computational Diffie-Hellman assumption) can still be produced.

that the message they receive is not simply a replay of an earlier execution of the protocol.

The advent of signcryption KEMs with outsider security removes one of these problems. If one uses a signcryption KEM in the same naive way that an encryption KEM is used above, then B can at least be assured that he is engaged in a protocol exchange with A as no other entity except B can forge encapsulations purporting to come from A. This only leaves the problem of freshness.

Generally, the problem of freshness can be solved either through the use of nonces or time-stamps. A nonce is a randomly generated number that is only ever used once for the purposes of authentication, whilst a time-stamp is a digital document that contains the date/time of its creation. A simple way of adding freshness to the naive method of key agreement we have been discussing is to send either a nonce or a time-stamp along with the encapsulation. The nonce/time-stamp must be integrally protected as it is sent; this could be achieved using a MAC computed using the newly agreed secret key. Hence, the complete key agreement mechanism using time-stamps would be:

1. A uses a signcryption KEM, along with B's public key and his own private key, to generate a symmetric key and an encapsulation of that key (K, C).
2. A uses the new key to compute a MAC τ of a time-stamp t_A under the key K, and sends C, t_A and τ to B.
3. B receives C, t_A and τ, and recovers the symmetric key K by running the decapsulation algorithm on C using A's public key and B's own private key.
4. B then checks that the time-stamp t_A is current and that the τ is a MAC of the time-stamp t_A. If either of these checks fail then B rejects the key K. Otherwise B accepts the key K.

The key agreement mechanism using nonces is similar:

1. B generates a random nonce r_B and sends this to A.
2. A uses a signcryption KEM, along with B's public key and his own private key, to generate a symmetric key and an encapsulation of that key (K, C).
3. A uses the new key to compute a MAC τ of a nonce r_B under the key K, and sends C and τ to B.
4. B receives C and τ, and recovers the symmetric key K by running the decapsulation algorithm on C using A's public key and B's own private key.
5. B then checks that τ is a MAC of the nonce r_B. If this check fails then B rejects the key K. Otherwise B accepts the key K.

These examples are very simple and suffer from several practical problems, for example, the schemes become weak if an attacker ever compromises a key K[5]. However, they do serve to show that secure key agreement mechanisms can

[5] If an attacker ever compromises a key K then they can force the user B to accept the key K as having come from A at any time in the future. This is achieved by resubmitting the encapsulation of K to B and computing the relevant MAC on the freshness value using the known key K.

be constructed from KEMs, but that signcryption KEMs with outsider security should be used rather than encryption KEMs. For more information about key agreement mechanisms, the reader is referred to Boyd and Mathuria [7].

8 Conclusions

We have shown that it is possible to create hybrid signcryption schemes with outsider security. The construction we have given is very similar to the KEM–DEM construction for hybrid encryption schemes, and, indeed, can even use the same DEM. Hence, any implementation that wishes to offer both encryption and signcryption can do so by implementing an encryption KEM, a signcryption KEM and a single DEM. There are two main advantages of this construction: the signcryption scheme it produces can be used to signcrypt messages of arbitrary length, and these schemes are often more efficient (particularly the verification-decryption algorithm) than more traditional schemes.

We have also presented the specification for ECISS–KEM — a simple, efficient and secure signcryption KEM whose security is based on the intractability of the computational Diffie-Hellman problem in large cyclic groups.

We note that this hybrid method of construction can *never* produce a signcryption scheme with insider security. Upon receipt of a signcryption (C_1, C_2), the receiver can forge the signcryption of any message m by recovering the symmetric key $K = \mathit{Decap}(pk_s, sk_r, C_1)$ and computing $C_2' = \textsc{Enc}_K(m)$. The ciphertext (C_1, C_2') is then a valid signcryption for the message m. Therefore, more complex constructions are needed to produce hybrid signcryption schemes with insider security.

Acknowledgements

The author would like to thank John Malone-Lee, Liqun Chen, Fred Piper, Bodo Möller, Yevgeniy Dodis and Stéphanie Alt for their helpful comments. The author would also like to thank several sets of anonymous reviewers for their comments. The author gratefully acknowledges the financial support of the EPSRC.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An encryption scheme based on the Diffie-Hellman problem. Submission to *P1363a: Standard Specifications for Public-Key Cryptography, Additional Techniques*, 2000.
2. J. H. An. Authenticated encryption in the public-key setting: Security notions and analyses. Available from `http://eprint.iacr.org/2001/079`, 2001.
3. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In L. Knudsen, editor, *Advances in Cryptology – Eurocrypt 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.

4. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.
5. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *Advances in Cryptology – Asiacrypt 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag, 2000.
6. M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In R. Bimal and W. Meier, editors, *Proceedings of the 11th Workshop on Fast Software Encryption (FSE 2004)*, volume 3017 of *Lecture Notes in Computer Science*, pages 391–408. Springer-Verlag, 2004.
7. C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.
8. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2004.
9. A. W. Dent. Hybrid signcryption schemes with insider security. Available from `http://www.isg.rhul.ac.uk/~alex/`, 2004.
10. Y. Dodis, M. J. Freedman, S. Jarecki, and S. Walfish. Optimal signcryption from any trapdoor permutation. Available from `http://eprint.iacr.org/2004/020/`, 2004.
11. International Organization for Standardization. *ISO/IEC 11770–3, Information technology — Security techniques — Key Management — Part 3: Mechanisms using asymmetric techniques*, 1999.
12. International Organization for Standardization. *ISO/IEC CD 18033–2, Information technology — Security techniques — Encryption Algorithms — Part 2: Asymmetric Ciphers*, 2003.
13. N. Koblitz and A. J. Menezes. Another look at "provable security". Available from `http://eprint.iacr.org/2004/152/`, 2004.
14. J. Malone-Lee. Signcryption with non-interactive non-repudiation. Technical Report CSTR-02-004, Department of Computer Science, University of Bristol, May 2004.
15. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *Proceedings of the Eighth ACM Conference on Computer and Communications Security (CCS-8)*, pages 196–205. ACM Press, 2001.
16. V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In B. Preneel, editor, *Advances in Cryptology – Eurocrypt 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 275–288. Springer-Verlag, 2000.
17. Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In B. Kaliski, editor, *Advances in Cryptology – Crypto '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.