Confidential Signatures and Deterministic Signcryption

Alexander W. Dent[1], Marc Fischlin[2], Mark Manulis[2], Martijn Stam[3], and Dominique Schröder[2]

[1] Royal Holloway, University of London, U.K.
[2] Darmstadt University of Technology, Germany
[3] LACAL, EPFL, Switzerland

Abstract. Encrypt-and-sign, where one encrypts and signs a message in parallel, is usually not recommended for confidential message transmission as the signature may leak information about the message. This motivates our investigation of confidential signature schemes, which hide all information about (high-entropy) input messages. In this work we provide a formal treatment of confidentiality for such schemes. We give constructions meeting our notions, both in the random oracle model and the standard model. As part of this we show that full domain hash signatures achieve a weaker level of confidentiality than Fiat-Shamir signatures. We then examine the connection of confidential signatures to signcryption schemes. We give formal security models for deterministic signcryption schemes for high-entropy and low-entropy messages, and prove encrypt-and-sign to be secure for confidential signature schemes and high-entropy messages. Finally, we show that one can derandomize any signcryption scheme in our model and obtain a secure deterministic scheme.

1 Introduction

A common mistake amongst novice cryptographers is to assume that digital signature schemes provide some kind of confidentiality service to the message being signed. The (faulty) argument in support of this statement is (a) that all signature schemes are of the "hash-and-sign" variety, which apply a hash function to a message before applying any kind of keyed operation, and (b) that a one-way hash function will hide all partial information about a message. Both facets of this argument are incorrect. However, it does suggest that notions of confidentiality for signature schemes are an interesting avenue of research.

The question of confidentiality of hash functions in signature schemes was previously considered by Canetti [7] as "content-concealing signatures"; however the original treatment only serves to motivate the concept of perfect one-way hash functions [7, 8]. We provide a more formal treatment here. The question of entropic security has been considered by several other authors. Dodis and Smith studied entropic secure primitives requiring that no function leaks their input [12]. Russell and Wang [22] consider the security of symmetric encryption schemes based on high-entropy messages, and several authors have considered the security of asymmetric encryption schemes based on high-entropy messages [3, 4, 6]. However, we are the first authors to consider the confidentiality of signatures and signcryption schemes in this scenario.

We believe that the concept of confidential signatures is intrinsically interesting and may prove to be useful in the construction of protocols in which two entities need to check that they are both aware of a particular message which (a) contains some confidential information, such as a password, and (b) contains a high entropy component, such as a confidential nonce.

Defining Confidential Signatures. Our first contribution is to define confidential signatures. Our starting point are high-entropy messages (signatures for messages with low entropy inevitably leak through the verification algorithm of the signature scheme). Our definitions are based on previous efforts for deterministic public-key encryption [3], and yield three models for confidential signature schemes:

- Weak confidentiality means that no information is leaked to a passive adversary, except possibly for information related to the technical details of the signature scheme.
- Mezzo confidentiality means that no information is leaked to a passive adversary (in possession of the verification key). Note that this is in contrast to deterministic public-key encryption where information cannot be hidden in such circumstances [3].
- Strong confidentiality means that no information is leaked to an active adversary (in possession of the verification key).

Our definitions are general enough to cover probabilistic and deterministic signature schemes, although we need an additional stipulation in the latter case, preventing the case where the leaked information is the unique signature itself.

Relation to Anonymous Signatures. There are similarities between confidential signatures and anonymous signatures [16, 23]. Anonymous signatures hide the identity of the signer of a high-entropy message, whereas confidential signatures hide all the information about the message itself. This is relationship between these two primitives is similar to the relationship between anonymous encryption and traditional public key encryption.

Constructing Confidential Signatures. We then show how to obtain confidential signatures. We first introduce the related concept of confidential hash functions, akin to hiding hash functions [3]. We prove that random oracles are confidential hash functions, as are perfectly one-way hash functions [7, 8] in a weaker form.

We then show that the use of weakly confidential hash functions in full domain hash (FDH) signature schemes yields weakly confidential signatures. We show that FDH signature schemes and Fiat-Shamir signatures are confidential in the random oracle model. We also show that strongly secure confidential signatures can be obtained in the standard model via the use of a randomness extractor [19, 20] (provided the message entropy lies above some fixed bound).

Applications to Signcryption. Secure message transmission is usually performed via the encrypt-then-sign paradigm, where the sender encrypts the message under the receiver's public encryption key and then signs the ciphertext with his own signing key. Signcryption schemes, introduced by [24], aim to gain efficiency by combining the two operations. One consequence of previous security definitions [1, 2] is that the encrypt-and-sign approach, where one encrypts the message and signs the message in parallel, does not provide a secure signcryption in general as the signature may reveal information about the message.

We introduce security notions for (possibly deterministic) signcryption schemes with high-entropy messages, along the lines of deterministic public-key encryption and confidential signatures. In case of signcryption schemes, we can also give a low-entropy-message version and show that this definition is strictly stronger than the definitions for high-entropy messages. We show that the parallelizable encrypt-and-sign scheme is high-entropy confidential if the underlying encryption scheme is IND-CCA2 and the signature scheme is confidential (and deterministic). We finally prove that we can derandomize any signcryption scheme to derive a secure deterministic scheme.

Besides the fact that some of our results require the signcryption scheme to be deterministic, we also believe that deterministic signcryption schemes may be intrinsically more secure than many current schemes. The reason is that most of the current signcryption schemes are based on discrete-logarithm-based digital signature schemes which are highly sensitive to imperfect randomness [18].

In situations where we have been forced due to size constraints to omit a theorem's proof, the proof can be found in the full version of the paper [10].

2 Confidential Signature Schemes

We formalise the notion of a confidential signature in three ways and give constructions. These confidentiality notions can be applied to either probabilistic or deterministic signature schemes.

2.1 Definition of Confidential Signature Schemes

A digital signature scheme is a tuple of efficient algorithms $\mathsf{SS} = (\mathtt{SS.Setup}, \mathtt{SS.Kg}, \mathtt{SS.Sign}, \mathtt{SS.Ver})$. All algorithms (in this article) are probabilistic polynomial-time (PPT) in the security parameter k (which we assume clear from the context). The parameter generation algorithm produces a set of parameters common to all users $\lambda_{ss} \overset{R}{\leftarrow} \mathtt{SS.Setup}(1^k)$; subsequently the key generation algorithm produces a public/private key pair $(pk, sk) \overset{R}{\leftarrow} \mathtt{SS.Kg}(\lambda_{ss})$. (Until Section 4.2 we will silently assume that λ_{ss} allows retrieval of k and both pk and sk allow retrieval of λ_{ss}, simplifying notation.) The signing algorithm takes a message $m \in \{0,1\}^*$ and the private key, and outputs a signature $\sigma \overset{R}{\leftarrow} \mathtt{SS.Sign}(sk, m)$. The verification algorithm takes as input a message, signature and public key, and outputs either a valid symbol $\top$ or an invalid symbol $\perp$. This is written $\mathtt{SS.Ver}(pk, m, \sigma)$.

```
$Expt_{\mathcal{A}}^{wSig-b}(k)$:
  $\lambda_{ss} \xleftarrow{R} \texttt{SS.Setup}(1^k)$
  $(pk, sk) \xleftarrow{R} \texttt{SS.Kg}(\lambda_{ss})$
  $(\boldsymbol{m}_0, t_0) \xleftarrow{R} \mathcal{A}_1(\lambda_{ss})$
  $(\boldsymbol{m}_1, t_1) \xleftarrow{R} \mathcal{A}_1(\lambda_{ss})$
  $\boldsymbol{\sigma}^* \leftarrow \texttt{SS.Sign}(sk, \boldsymbol{m}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2^{\texttt{SS.Sign}(sk,\cdot)}(pk, \boldsymbol{\sigma}^*)$
  If $t' = t_0$ then output 1
  Else return 0
```

```
$Expt_{\mathcal{A}}^{mSig-b}(k)$:
  $\lambda_{ss} \xleftarrow{R} \texttt{SS.Setup}(1^k)$
  $(pk, sk) \xleftarrow{R} \texttt{SS.Kg}(\lambda_{ss})$
  $(\boldsymbol{m}_0, t_0) \xleftarrow{R} \mathcal{A}_1(pk)$
  $(\boldsymbol{m}_1, t_1) \xleftarrow{R} \mathcal{A}_1(pk)$
  $\boldsymbol{\sigma}^* \leftarrow \texttt{SS.Sign}(sk, \boldsymbol{m}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2^{\texttt{SS.Sign}(sk,\cdot)}(pk, \boldsymbol{\sigma}^*)$
  If $t' = t_0$ then output 1
  Else return 0
```

```
$Expt_{\mathcal{A}}^{sSig-b}(k)$:
  $\lambda_{ss} \xleftarrow{R} \texttt{SS.Setup}(1^k)$
  $(pk, sk) \xleftarrow{R} \texttt{SS.Kg}(\lambda_{ss})$
  $(\boldsymbol{m}_0, t_0) \xleftarrow{R} \mathcal{A}_1^{\texttt{SS.Sign}(sk,\cdot)}(pk)$
  $(\boldsymbol{m}_1, t_1) \xleftarrow{R} \mathcal{A}_1^{\texttt{SS.Sign}(sk,\cdot)}(pk)$
  $\boldsymbol{\sigma}^* \leftarrow \texttt{SS.Sign}(sk, \boldsymbol{m}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2^{\texttt{SS.Sign}(sk,\cdot)}(pk, \boldsymbol{\sigma}^*)$
  If $t' = t_0$ then output 1
  Else return 0
```

Fig. 1. Notions of confidentiality for (a) weakly confidential signature schemes; (b) mezzo confidential signature schemes; (c) strongly confidential signature schemes. The signing algorithm is applied to the message vector $\boldsymbol{m}$ component-wise.

The standard notion for signature security is that of unforgeability under chosen message attacks (see Appendix A.1 for formal definitions).

We present three confidentiality notions for a digital signature scheme — see Figure 1. These notions are split depending on the adversary's capabilities, which corresponds in a natural way to real-life scenarios where it may be possible to derive some information about a message from a signature which might be deemed practically useless, e.g., the value of the hash of the message, but leakage of which cannot be avoided.

In the weak confidentiality model, the attacker should not be able to determine any information about the messages apart from that which can be obtained directly from the signature itself. Mezzo confidentiality models the scenario where the attacker is able to retrieve public keys of the users, but cannot interact directly with their communication network and obtain signatures of messages. In the strong model, an active attacker should not be able to determine any information about the messages apart from the signature itself.

For $x \in \{w, m, s\}$, the attacker $\mathcal{A}$'s advantage in the $xSig$ game is defined to be:

$$Adv_{\mathcal{A}}^{xSig}(k) = |\Pr[Expt_{\mathcal{A}}^{xSig\text{-}0}(k) = 1] - \Pr[Expt_{\mathcal{A}}^{xSig\text{-}1}(k) = 1]| .$$

A signature scheme is weakly confidential (resp. mezzo confidential/strongly confidential) if all PPT attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ have negligible advantage $Adv_{\mathcal{A}}^{xSig}(k)$ in the *wSig* (resp. *mSig*/*sSig*) security game, subject to the following restraints:

- Pattern preserving: there exist a length function $\ell(k)$ and equality functions $\diamond_{ij} \in \{=, \neq\}$ $(1 \leq i, j \leq \ell(k))$ such that for any admissible input a in the corresponding game and all possible $(\boldsymbol{m}, t) \xleftarrow{R} \mathcal{A}_1(a)$ we have that $|\boldsymbol{m}| = \ell(k)$ and $m_i \diamond_{ij} m_j$.
- High entropy: the function $\pi(k) = \max_{m \in \{0,1\}^*} \Pr[m_i = m : (\boldsymbol{m}, t) \xleftarrow{R} \mathcal{A}_1(a)]$ is negligible, where the probability is over $\mathcal{A}_1$'s random tape only (and $i \in \mathbb{N}$ and all choices of the other algorithms are fixed). The value $\mu(k) = -\log_2 \pi(k)$ is termed the adversary's *min entropy*.

For deterministic schemes we need the following additional constraint, ruling out trivial attacks:

```
$\texttt{SS.Kg}'(\lambda_{ss})$:                 $\texttt{SS.Sign}'(sk\|r, m)$:                 $\texttt{SS.Ver}'(pk\|r, m, \sigma)$:
  $r \xleftarrow{R} \{0,1\}^k$                   If $m = m'\|r$                             If $m = m'\|r$
  $(pk, sk) \xleftarrow{R} \texttt{SS.Kg}(\lambda_{ss})$        Return $\texttt{SS.Sign}(sk, m)\|m$         Parse $\sigma$ as $\sigma'\|m$
  Return $(pk\|r, sk\|r)$                       Else                                            $\sigma \leftarrow \sigma'$
                                                    Return $\texttt{SS.Sign}(sk, m)$        Return $\texttt{SS.Ver}(pk, m, \sigma)$
```

Fig. 2. A signature scheme which is weakly confidential but not mezzo confidential.

– Signature free: $\mathcal{A}_1$ does not output a message $m_i \in \boldsymbol{m}$ where it has queried the signature oracle on m_i. (This security requirement only affects strongly confidential signature schemes.)

The latter condition prevents an attacker against a deterministic scheme from "winning" by setting $t \leftarrow \texttt{SS.Sign}(sk, m)$ — i.e., it prevents the attacker from "winning" the game simply by determining that the message m has the property that its unique signature is $\texttt{SS.Sign}(sk, m)$.

The notions of confidentiality are strictly increasing in strength. If SS is a weakly confidential signature schemes, then Figure 2 depicts a scheme which is weakly confidential but not mezzo confidential. Similarly, if SS is a mezzo confidential signature scheme, then Figure 3 shows a scheme which is mezzo confidential but not strongly confidential.

```
$\texttt{SS.Kg}'(\lambda_{ss})$:
  $(pk, sk) \xleftarrow{R} \texttt{SS.Kg}(\lambda_{ss})$
  $r \xleftarrow{R} \{0,1\}^k$
  $\sigma_r \leftarrow \texttt{SS.Sign}(sk, 0\|r)$
  Return $(pk, sk\|r\|\sigma_r)$

$\texttt{SS.Sign}'(sk\|r\|\sigma_r, m)$:
  If $m = m'\|r\|\sigma_r$
      Set $\sigma' \leftarrow \texttt{SS.Sign}(sk, 1\|m)$
      Return $\sigma = (\sigma', m)$
  Else
      Set $\sigma \leftarrow \texttt{SS.Sign}(sk, 2\|m)$
      Return $\sigma = (\sigma', r, \sigma_r)$

$\texttt{SS.Ver}'(pk, m, \sigma)$:
  If $\sigma = (\sigma', m')$
      Parse $m'$ as $m' = m''\|r'\|\sigma'_r$
      Return $\top$ iff
          $\texttt{SS.Ver}(pk, 1\|m', \sigma') = \top$, and
          $m = m'$, and
          $\texttt{SS.Ver}(pk, 0\|r', \sigma'_r) = \top$
  If $\sigma = (\sigma', r', \sigma'_r)$
      Return $\top$ iff
          $\texttt{SS.Ver}(pk, 2\|m, \sigma') = \top$, and
          $m \neq m''\|r'\|\sigma'_r$ for any $m'' \in \{0,1\}^*$,
          and $\texttt{SS.Ver}(pk, 0\|r', \sigma'_r) = \top$
  Else return $\perp$
```

Fig. 3. A signature scheme which is mezzo confidential but not strongly confidential.

3 Confidential Hash Functions and Signature Schemes

3.1 Confidential Hash Functions

We recap the notion of a *hiding* hash function by Bellare *et al.* [3], but call such functions confidential here. For our purposes, a hash function $\mathsf{H} = (\texttt{H.Kg}, \texttt{H})$ is

```
$Expt_{\mathcal{A}}^{\text{wHash-b}}(k)$:
  $\mathtt{H} \xleftarrow{R} \mathtt{H.Kg}(1^k)$
  $(\boldsymbol{x}_0, t_0) \xleftarrow{R} \mathcal{A}_1(1^k)$
  $(\boldsymbol{x}_1, t_1) \xleftarrow{R} \mathcal{A}_1(1^k)$
  $\boldsymbol{h} \leftarrow \mathtt{H}(\boldsymbol{x}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2(\mathtt{H}, \boldsymbol{h})$
  If $t' = t_0$ then output 1
  Else return 0
```

```
$Expt_{\mathcal{A}}^{\text{sHash-b}}(k)$:
  $\mathtt{H} \xleftarrow{R} \mathtt{H.Kg}(1^k)$
  $(\boldsymbol{x}_0, t_0) \xleftarrow{R} \mathcal{A}_1(\mathtt{H})$
  $(\boldsymbol{x}_1, t_1) \xleftarrow{R} \mathcal{A}_1(\mathtt{H})$
  $\boldsymbol{h} \leftarrow \mathtt{H}(\boldsymbol{x}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2(\mathtt{H}, \boldsymbol{h})$
  If $t' = t_0$ then output 1
  Else return 0
```

Fig. 4. Notions of confidentiality for (a) weakly confidential hash functions; (b) strongly confidential hash functions. The hash function is applied to the data vector $\boldsymbol{x}$ component-wise.

a PPT pair of algorithms for key generation and hashing, respectively. We will identify the description output by the key generation algorithm $\mathtt{H.Kg}$ with the hash function $\mathtt{H}$ itself. The collision-finding advantage $Adv_{\mathcal{A}}^{col}$ of an attacker $\mathcal{A}$ against a hash function H is defined as

$$Adv_{\mathsf{H},\mathcal{A}}^{col}(k) = \Pr\left[\begin{array}{c}\mathtt{H}(x;r) = \mathtt{H}(x';r') \\ \text{and } (x,r) \neq (x',r')\end{array} : (x, x', r, r') \xleftarrow{R} \mathcal{A}(\mathtt{H});\ \mathtt{H} \xleftarrow{R} \mathtt{H.Kg}(1^k)\right].$$

The hash function H is called *collision-resistant* if all PPT attackers $\mathcal{A}$ have negligible advantage $Adv_{\mathsf{H},\mathcal{A}}^{col}(k)$ (as a function of k). We require that the hash function is hiding/confidential against an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ playing one of the games in Figure 4. For $x \in \{w, s\}$ the attacker's advantage is defined to be

$$Adv_{\mathsf{H},\mathcal{A}}^{\text{xHash}}(k) = \left|\Pr[Expt_{\mathcal{A}}^{\text{xHash-0}}(k) = 1] - \Pr[Expt_{\mathcal{A}}^{\text{xHash-1}}(k) = 1]\right|.$$

A hash function is *weakly (resp. strongly) confidential* if every PPT attacker $\mathcal{A}$ has negligible advantage in the corresponding game subject to the following restraints:

- Pattern preserving: there exist a length function $\ell(k)$ and equality functions $\diamond_{ij} \in \{=, \neq\}$ $(1 \leq i, j \leq \ell(k))$ such that for all possible $(\boldsymbol{x}, t) \xleftarrow{R} \mathcal{A}_1(1^k)$ we have that $|\boldsymbol{x}| = \ell(k)$ and $x_i \diamond_{ij} x_j$.
- High entropy: the function $\pi(k) = \max_{x \in \{0,1\}^*} \Pr[x_i = x : (\boldsymbol{x}, t) \xleftarrow{R} \mathcal{A}_1(a)]$ is negligible where the probability is only over $\mathcal{A}_1$'s random tape. We define $\mu(k) = -\log_2 \pi(k)$ to be the adversary's *minimum entropy.*

Note that collision-resistant deterministic hash functions cannot achieve strong confidentiality because an adversary $\mathcal{A}_1$ can set $t = \mathtt{H}(x)$ for some message x and $\mathcal{A}_2$ can easily obtain this value from the hash vector $\boldsymbol{h}$. We also note that for "unkeyed" hash functions both notions are equivalent and so no unkeyed, deterministic hash function can be considered confidential (unless the hash function is almost constant).

In the random oracle model, where the adversary is granted oracle access to the hash function $\mathtt{H}$ instead of receiving the description as input, we give

$\mathcal{A}_1$ access to the random oracle in the strong case, but deny $\mathcal{A}_1$ access to $\mathtt{H}$ in the weak case. It is easy to see that a random oracle thus achieves weak confidentiality, whereas the above attack on deterministic functions still applies in the strong case. However, under the additional constraint that $\mathcal{A}_1$ does not query $\mathtt{H}$ about any x in its output $\boldsymbol{x}$ (*hash-free adversaries*) a random oracle is also strongly confidential:

Proposition 1 (Confidentiality of Random Oracles). *For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ where $\mathcal{A}_1$ outputs vectors of length $\ell(k)$ and with min-entropy $\mu(k) = -\log \pi(k)$, and where $\mathcal{A}_2$ makes at most $q_h(k)$ queries to the random oracle, we have*

$$Adv^{xHash}_{\mathtt{H},\mathcal{A}}(k) \leq 2 \cdot q_h(k) \cdot \ell(k) \cdot \pi(k)$$

for $x \in \{w, s\}$ where $\mathcal{A}$ is assumed to be hash-free (in the strong case).

As for constructions in the standard model, we note that perfectly one-way functions (POWs) [7, 8] provide a partial solution. POWs have been designed to hide all information about preimages, akin to our confidentiality notion. However, all known constructions of POWs are only good for fixed (sets of) input distributions where the distributions can depend only on the security parameter but not the hash function description. Furthermore, known POWs usually require the conditional entropy of any x_i to be high, given the other x_j's. In light of this, any $\ell(k)$-valued perfectly one-way function [8] is a weakly confidential hash function. Hence, we can build such hash functions based, for example, on claw-free permutations [8] or one-way permutations [8, 15].

3.2 Full-Domain Hash Signatures

A *full-domain hash (FDH) signature scheme* FDH *for deterministic hash function* H is a signature scheme in which the signing algorithm computes a signature as $\sigma = f(\mathtt{H}(m))$ for some secret function f, and the verification algorithm checks that $g(\sigma) = \mathtt{H}(m)$ for some public function g. More formally (assuming that $\mathtt{FDH.Setup}(1^k)$ outputs $\lambda_{ss} = 1^k$ and that there exists a PPT algorithm which generates the functions $(f, g) \leftarrow \mathtt{FDH.Kg}'(\lambda_{ss})$):

```
$\mathtt{FDH.Kg}(\lambda_{ss})$:
  $(f, g) \leftarrow \mathtt{FDH.Kg}'(\lambda_{ss})$
  $\mathtt{H} \leftarrow \mathtt{H.Kg}(1^k)$
  $(pk, sk) = ((g, \mathtt{H}), (f, \mathtt{H}))$
  Return $(pk, sk)$

$\mathtt{FDH.Sign}(sk, m)$:
  Parse $sk$ as $(f, \mathtt{H})$
  Return $\sigma = f(\mathtt{H}(m))$

$\mathtt{FDH.Ver}(pk, m, \sigma)$:
  Parse $pk$ as $(g, \mathtt{H})$
  Return $\top$ if $\mathtt{H}(m) = g(\sigma)$
  Otherwise return $\bot$
```

Unforgeability of FDH signatures in the ROM has been shown in [5, 9].

Proposition 2 (Weak Confidentiality of FDH). *The FDH-signature scheme* FDH *for hash function* H *is weakly confidential if* H *is weakly confidential. More precisely, for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the weak confidentiality of* FDH*, where $\mathcal{A}_1$ outputs $\ell(k)$ messages and $\mathcal{A}_2$ makes at most $q_s(k)$ signature* *queries, there exists an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the weak confidentiality of the hash function such that*

$$Adv^{wSig}_{\mathsf{FDH},\mathcal{A}}(k) \leq Adv^{wHash}_{\mathsf{H},\mathcal{B}}(k),$$

where $\mathcal{B}_1$'s running time is identical to the one of $\mathcal{A}_1$, and $\mathcal{B}_2$'s running time is the one of $\mathcal{A}_2$ plus $Time_{\texttt{FDH.Kg}}(k) + (q_s + \ell(k)) \cdot Time_{\texttt{FDH.Sign}}(k) + O(k)$.

The proof actually shows that the signature scheme remains confidential for an adversarially chosen key pair (f, g), i.e., confidentiality only relies on the confidentiality of the hash function. Moreover, by Proposition 1, we have that FDH-signature schemes are weakly confidential in the random oracle model.

Proof. Assume that FDH is not weakly confidential and that there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ successfully breaking this property. Then we construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the weak confidentiality of the hash function as follows. Adversary $\mathcal{B}_1$ on input 1^k runs $\mathcal{A}_1$ on input 1^k and outputs this algorithm's answer $(\boldsymbol{m}, t)$.

Algorithm $\mathcal{B}_2$ receives as input a description $\texttt{H}$ of the confidential hash function and a vector $\boldsymbol{h}$ of hash values. $\mathcal{B}_2$ runs $(f, g) \leftarrow \texttt{FDH.Kg}'(1^k)$, sets $pk \leftarrow (g, \texttt{H})$ and $sk \leftarrow (f, \texttt{H})$, and computes signatures $\boldsymbol{\sigma^*} = f(\boldsymbol{h})$. It invokes $\mathcal{A}_2$ on $(1^k, pk, \boldsymbol{\sigma^*})$ and answers each subsequent signature request for message m by computing $\sigma = \texttt{FDH.Sign}(sk, m)$. When $\mathcal{A}_2$ outputs t' algorithm $\mathcal{B}_2$ copies this output and stops.

It is easy to see that $\mathcal{B}$'s advantage attacking the confidentiality of the hash function is identical to $\mathcal{A}$'s advantage attacking the confidentiality of the FDH signature scheme (the fact that $\mathcal{A}_1$ preserves pattern and produces high-entropy messages carries over to $\mathcal{B}_1$). □

No (unforgeable) FDH-signature scheme is mezzo confidential, because a signature on the message m leaks the value $\texttt{H}(m)$. More formally, an attacker $\mathcal{A}_1$ can pick a message $m \overset{R}{\leftarrow} \{0,1\}^k$ and set $t \leftarrow \texttt{H}(m)$. Adversary $\mathcal{A}_2$ then receives $\sigma \leftarrow f(\texttt{H}(m))$ and can recover $t = \texttt{H}(m)$ by computing $g(\sigma)$.

3.3 Strongly Confidential Signatures in the ROM

Recall from the previous section that FDH signatures leak the hash value of a message. To prevent this, we make the hashing process probabilistic and compute $(r, \texttt{H}(r, m))$ for randomness r. Then $\mathcal{A}_1$ cannot predict the hash values of the challenge messages due to r (which becomes public only afterwards) and $\mathcal{A}_2$ cannot guess the hash values due to the entropy in the message m (even though r is then known). Our instantiation is shown in Figure 5.

Proposition 3 (Random Oracle Instantiation). *If $\texttt{H}$ is a hash function modeled as a random oracle, then the signature scheme SS' is strongly confidential. That is, for any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the strong confidentiality of the signature scheme SS', where $\mathcal{A}_1$ outputs a vector of length $\ell(k)$ and with*

Suppose $\mathsf{SS} = (\texttt{SS.Setup}, \texttt{SS.Kg}, \texttt{SS.Sign}, \texttt{SS.Ver})$ is a signature scheme. We define a new signature scheme SS' as follows (where $\texttt{SS.Setup}' \equiv \texttt{SS.Setup}$):

```
$\texttt{SS.Kg}'(\lambda_{ss})$:
  $(pk, sk) \leftarrow \texttt{SS.Kg}(\lambda_{ss})$
  $\texttt{H} \overset{R}{\leftarrow} \texttt{H.Kg}(1^k)$
  $pk' \leftarrow (pk, \texttt{H})$; $sk' \leftarrow (sk, \texttt{H})$
  Return $(pk', sk')$
```

```
$\texttt{SS.Sign}'(sk', m)$:
  Parse $sk'$ as $(sk, \texttt{H})$
  $r \overset{R}{\leftarrow} \{0,1\}^k$
  $h \leftarrow \texttt{H}(r, m)$
  $\sigma' \leftarrow \texttt{SS.Sign}(sk, h)$
  $\sigma \leftarrow (\sigma', r)$
  Return $\sigma$
```

```
$\texttt{SS.Ver}'(pk', m, \sigma)$:
  Parse $pk'$ as $(pk, \texttt{H})$
  Parse $\sigma$ as $(\sigma', r)$
  Return $\texttt{SS.Ver}(pk, \texttt{H}(r, m), \sigma')$
```

Fig. 5. Construction of a strongly confidential signature scheme in the ROM.

min-entropy $\mu(k) = -\log \pi(k)$, and where $\mathcal{A}_2$ asks at most q_h oracle queries (signing queries and direct hash oracle queries), we have

$$Adv^{sSig}_{\mathsf{SS}',\mathcal{A}}(k) \leq 2 \cdot q_h(k) \cdot \ell(k) \cdot (2^{-k} + \pi(k))\,.$$

Clearly, the scheme is also (strongly) unforgeable if the underlying signature scheme is (strongly) unforgeable.

3.4 Fiat-Shamir Signature Schemes

Our second instantiation is based upon the Fiat-Shamir paradigm [14] that turns every (three-round) identification scheme into a signature scheme. An identification scheme (ID scheme) is defined by a triplet (G, S, R), where G is a key generation algorithm and the sender S wishes to prove his identity to the receiver R. More formally: $G(1^k)$ is an efficient algorithm that outputs a key pair (ipk, isk). $(S(isk), R(ipk))$ are interactive algorithms and it is required that $\Pr[\,(S(isk), R(ipk)) = 1] = 1$ (where the probability is taken over the coin tosses of S, R and G). A canonical ID scheme is a 3-round ID scheme $(\alpha; \beta; \gamma)$ in which α is sent by the sender S, β by the receiver R and consists of R's random coins, and γ is sent by the sender. For a sender S with randomness r, we denote $\alpha = S(isk; r)$ and $\gamma = S(isk, \alpha, \beta; r)$. The Fiat-Shamir signature scheme is given in Figure 6.

In order to prove the confidentiality of this scheme, we need to assume that the commitment α of the Fiat-Shamir scheme has non-trivial entropy. This can always be achieved by appending public randomness.

Proposition 4 (Fiat-Shamir Instantiation). *If $\texttt{H}$ is a hash function modeled as a random oracle, then the Fiat-Shamir instantiation SS'' for non-trivial commitments is strongly confidential. More precisely, for any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the strong confidentiality of the signature scheme SS'' where $\mathcal{A}_1$ outputs a message vector of length $\ell(k)$ with min-entropy $\mu(k) = -\log \pi(k)$, α has min-entropy $\mu'(k) = -\log \pi'(k)$, and $\mathcal{A}_2$ asks at most q_h oracle queries (signing queries and direct hash oracle queries), we have*

$$Adv^{sSig}_{\mathsf{SS}'',\mathcal{A}}(k) \leq 2 \cdot q_h(k) \cdot \ell(k) \cdot (\pi(k) + \pi'(k)).$$

Suppose (G, S, R) is a canonical identification scheme and H is a hash function family. We define the signature scheme $\mathsf{SS}'' = (\mathtt{SS.Setup}'', \mathtt{SS.Kg}'', \mathtt{SS.Sign}'', \mathtt{SS.Ver}'')$ as follows (where $\mathtt{SS.Setup}(1^\lambda)$ returns $\lambda_{ss} = 1^\lambda$):

```
$\mathtt{SS.Kg}''(\lambda_{ss})$:
  $(ipk, isk) \leftarrow G(\lambda_{ss})$
  $\mathtt{H} \overset{R}{\leftarrow} \mathtt{H.Kg}(1^k)$
  $pk' \leftarrow (ipk, \mathtt{H})$; $sk' \leftarrow (isk, \mathtt{H})$
  Return $(pk', sk')$

$\mathtt{SS.Sign}''(sk', m)$:
  Parse $sk'$ as $(isk, \mathtt{H})$
  $r \overset{R}{\leftarrow} \{0,1\}^k$
  $\alpha \leftarrow S(isk; r)$
  $\beta \leftarrow \mathtt{H}(\alpha, m)$
  $\gamma \leftarrow S(isk, \alpha, \beta; r)$
  $\sigma \leftarrow (\alpha, \beta, \gamma)$
  Return $\sigma$

$\mathtt{SS.Ver}''(pk', m, \sigma)$:
  Parse $pk'$ as $(ipk, \mathtt{H})$
  Parse $\sigma$ as $(\alpha, \beta, \gamma)$
  $\beta' \leftarrow \mathtt{H}(\alpha, m)$
  Return 1 iff $\beta = \beta'$
      and $R(ipk, \alpha, \beta, \gamma) = 1$
```

Fig. 6. The Fiat-Shamir paradigm that turns every ID scheme into a signature scheme.

3.5 Strongly Confidential Signatures from Randomness Extraction

Our instantiation in the standard model relies on randomness extractors [19, 20] and is depicted in Figure 7. The main idea is to smooth the distribution of the message via an extractor, and to sign the almost uniform value h.

Recall that a strong (a, b, n, t, ϵ)-extractor is an efficient algorithm $\mathsf{Ext} : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^n$ which takes some random input $m \in \{0,1\}^a$ (sampled according to some distribution with min-entropy at least t) and some randomness $r \in \{0,1\}^b$. It outputs $h \leftarrow \mathsf{Ext}(m, r)$ such that the statistical distance between (r, h) and (r, u) is at most ϵ for uniform random values $r \in \{0,1\}^b$ and $u \in \{0,1\}^n$.

To ensure unforgeability we need to augment the extractor's extraction property by collision-resistance, imposing the requirement that the extractors be keyed and introducing dependency of the extractor's parameters a, b, n, t, ϵ on the security parameter k. For a survey about very efficient constructions of such collision-resistant extractors see [11].

In order to use extractors, we need a stronger assumption on the message distribution: we assume that the adversary $\mathcal{A}_1$ now outputs vectors of messages such that each message in the vector has min-entropy greater than some fixed bound $\mu(k)$ *given the other messages.* Observe that the collision-resistance requirement on the extractor implies that μ must be super-logarithmic. We say that the output has *conditional min-entropy* $\mu(k)$.

Proposition 5 (Extractor Instantiation). *If* Ext *is an* (a, b, n, t, ϵ)*-extractor then the extractor instantiation of* SS''' *is strongly confidential. More specifically, for any attacker* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against the strong confidentiality of the signature scheme* SS'''*, where* $\mathcal{A}_1$ *outputs a vector of length* $\ell(k)$ *with conditional min-entropy* $\mu(k) \geq t(k)$*, we have*

$$Adv^{sSig}_{\mathsf{SS}''',\mathcal{A}}(k) \leq 2 \cdot \ell(k) \cdot \epsilon(k).$$

Note that our construction of the randomness extractor operates on messages of a fixed length of $a(k)$ input bits, and the signature length depends on this

Suppose $\mathsf{SS} = (\texttt{SS.Setup}, \texttt{SS.Kg}, \texttt{SS.Sign}, \texttt{SS.Ver})$ is a signature scheme. We define a new signature scheme SS''' as follows (where $\texttt{SS.Setup}''' \equiv \texttt{SS.Setup}$):

```
$\texttt{SS.Kg}'''(\lambda_{ss})$:
    $(pk, sk) \leftarrow \texttt{SS.Kg}(\lambda_{ss})$
    Choose an extractor $\mathsf{Ext}$
    $pk' \leftarrow (pk, \mathsf{Ext})$
    $sk' \leftarrow (sk, \mathsf{Ext})$
    Return $(pk', sk')$

$\texttt{SS.Sign}'''(sk', m)$:
    Parse $sk'$ as $(sk, \mathsf{Ext})$
    $r \overset{R}{\leftarrow} \{0,1\}^b$
    $h \leftarrow \mathsf{Ext}(m, r)$
    $\sigma' \leftarrow \texttt{SS.Sign}(sk, h)$
    $\sigma \leftarrow (\sigma', r)$
    Return $\sigma$

$\texttt{SS.Ver}'''(pk', m, \sigma)$:
    Parse $pk'$ as $(pk, \mathsf{Ext})$
    Parse $\sigma$ as $(r, \sigma')$
    Set $h \leftarrow \mathsf{Ext}(m, r)$
    Return $\texttt{SS.Ver}(pk, h, \sigma')$
```

Fig. 7. Construction of strongly confidential signature scheme based on randomness extractors.

value $a(k)$. To process larger messages we can first hash input messages with a collision-resistant hash function, before passing it to the extractor. In this case, some care must be taken to determine a correct bound for the entropy lost through the hash function computation.

4 Deterministic Signcryption

Signcryption is a public-key primitive which aims to simultaneously provide message confidentiality and message integrity. Signcryption was introduced by Zheng [24] and security models were independently introduced by An, Dodis and Rabin [1] and by Baek, Steinfeld and Zheng [2]. Similar to public-key encryption, achieving confidentiality in the formal security models requires that signcryption is a randomised process; however, we may also consider the confidentiality of deterministic signcryption schemes on high-entropy message spaces. We will also see that a practical version of confidentiality may even be achieved by a deterministic signcryption scheme for low entropy message distributions.

4.1 Notions of Confidentiality for Signcryption Schemes

A signcryption scheme is a tuple of PPT algorithms $\mathsf{SC} = (\texttt{SC.Setup}, \texttt{SC.Kg}_\texttt{s}, \texttt{SC.Kg}_\texttt{r}, \texttt{SC.SignCrypt}, \texttt{SC.UnSignCrypt})$. The setup algorithm generates public parameters $\lambda_{sc} \overset{R}{\leftarrow} \texttt{SC.Setup}(1^k)$ common to all algorithms. We will generally assume that all algorithms take λ_{sc} as an implicit input, even if it is not explicitly stated. The sender key-generation algorithm generates a key pair for the sender $(pk_S, sk_S) \overset{R}{\leftarrow} \texttt{SC.Kg}_\texttt{s}(\lambda_{sc})$ and the receiver key-generation algorithm generates a key pair for a receiver $(pk_R, sk_R) \overset{R}{\leftarrow} \texttt{SC.Kg}_\texttt{r}(\lambda_{sc})$. The signcryption algorithm takes as input a message $m \in \mathcal{M}$, the sender's private key sk_S, and the receiver's public key pk_R, and outputs a signcryption ciphertext $C \overset{R}{\leftarrow} \texttt{SC.SignCrypt}(sk_S, pk_R, m)$. The unsigncryption algorithm takes as input a ciphertext $C \in \mathcal{C}$, the sender's public key pk_S, and the receiver's private key sk_R, and outputs either a message $m \overset{R}{\leftarrow} \texttt{SC.UnSignCrypt}(pk_S, sk_R, C)$ or an error symbol $\perp$.

It is interesting to consider the basic attack on a deterministic signcryption scheme. In such an attack, the attacker picks two messages (m_0, m_1) and receives a signcryption C^* of the message m_b. The attacker checks whether C^* is the signcryption of m_0 by requesting the signcryption of m_0 from the signcryption oracle. As in the case of public-key encryption, we may prevent this basic attack by using a high-entropy message space and so prevent the attacker being able to determine which message to query to the signcryption oracle. However, unlike the case of public-key encryption, we may also prevent this attacker by forbidding the attacker to query the signcryption oracle on m_0 and m_1. We can therefore differentiate between the high-entropy case (in which the message distribution chosen by the attacker has high entropy) and the low-entropy case (in which the attacker is forbidden from querying the signcryption oracle on a challenge message).

We give definitions for the high-entropy and low-entropy confidentiality in Figure 8. In both cases, i.e. for $x \in \{h, l\}$, the attacker's advantage is defined as

$$Adv^{xSCR}_{\mathsf{SS},\mathcal{A}}(k) = |\Pr[Expt^{xSCR-1}_{\mathcal{A}} = 1] - \Pr[Expt^{xSCR-0}_{\mathcal{A}} = 1]| \, .$$

A signcryption scheme is high-entropy confidential if every PPT attacker $\mathcal{A}$ has negligible advantage in the hSCR game subject to the following restrictions:

– Strongly pattern preserving: there exists a length function $\ell(k)$, message length functions $q_i(k)$, and equality functions $\diamond_{ij} \in \{=, \neq\}$ $(1 \leq i, j \leq \ell(k))$ such that for all possible $(\boldsymbol{m}, t) \overset{R}{\leftarrow} \mathcal{A}_1(\lambda_{sc}, pk^*_S, pk^*_R)$ we have that $|\boldsymbol{m}| = \ell(k)$, $|m_i| = q_i(k)$ and $m_i \diamond_{ij} m_j$.
– High entropy: the function $\pi(k) = \max_{m \in \{0,1\}^*} \Pr[m_i = m : (\boldsymbol{m}, t) \overset{R}{\leftarrow} \mathcal{A}_1(a)]$ is negligible where the probability is only over $\mathcal{A}_1$'s random tape. The value $\mu(k) = -\log \pi(k)$ is known as the adversary's minimum entropy.
– Signature free: $\mathcal{A}_1$ does not output a message $m_i \in \boldsymbol{m}$ where it has queried the signcryption oracle on the pair (pk^*_R, m_i).
– Non-trivial: $\mathcal{A}_2$ does not query the unsigncryption oracle on any pair (pk^*_S, C) where $C \in \boldsymbol{C}^*$.

A signcryption scheme is low-entropy confidential if every PPT attacker $\mathcal{A}$ has negligible advantage in the lSCR game subject to the restrictions that $\mathcal{A}$ never queries the encryption oracle on either (pk^*_R, m_0) or (pk^*_R, m_1), and $\mathcal{A}_2$ never queries the decryption oracle on (pk^*_S, C^*).

Proposition 6. *Any deterministic signcryption scheme* SC *which is low-entropy confidential is also high-entropy confidential. In particular, for any adversary* $\mathcal{A}$ *against high-entropy confidentiality, making at most* $q_s(k)$ *signcryption queries and where* $\mathcal{A}_1$ *outputs* $\ell(k)$ *messages with min-entropy* $\mu(k) = -\log \pi(k)$*, there exists an adversary* $\bar{\mathcal{A}}$ *such that*

$$Adv^{hSCR}_{\mathsf{SC},\mathcal{A}}(k) \leq \ell(k) \cdot Adv^{lSCR}_{\mathsf{SC},\bar{\mathcal{A}}}(k) + 4 \cdot q_s(k) \cdot \ell(k) \cdot \pi(k),$$

where the running time of $\bar{\mathcal{A}}$ *equals the time of* $\mathcal{A}$ *plus* $O(k)$*.*

```
$Expt_{\mathcal{A}}^{hSCR-b}(k)$:
  $\lambda_{sc} \xleftarrow{R} \texttt{SC.Setup}(1^k)$
  $(pk_S^*, sk_S^*) \xleftarrow{R} \texttt{SC.Kg}_\texttt{s}(\lambda_{sc})$
  $(pk_R^*, sk_R^*) \xleftarrow{R} \texttt{SC.Kg}_\texttt{r}(\lambda_{sc})$
  $(\boldsymbol{m}_0, t_0) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}}(\lambda_{sc}, pk_S^*, pk_R^*)$
  $(\boldsymbol{m}_1, t_1) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}}(\lambda_{sc}, pk_S^*, pk_R^*)$
  $\boldsymbol{C}^* \leftarrow \texttt{SC.SignCrypt}(\lambda_{sc}, sk_S^*, pk_R^*, \boldsymbol{m}_b)$
  $t' \xleftarrow{R} \mathcal{A}_2^{\mathcal{O}}(\lambda_{sc}, pk_S^*, pk_R^*, \boldsymbol{C}^*)$
  If $t' = t_0$ then output 1
  Else return 0
```

```
$Expt_{\mathcal{A}}^{lSCR-b}(k)$:
  $\lambda_{sc} \xleftarrow{R} \texttt{SC.Setup}(1^k)$
  $(pk_S^*, sk_S^*) \xleftarrow{R} \texttt{SC.Kg}_\texttt{s}(\lambda_{sc})$
  $(pk_R^*, sk_R^*) \xleftarrow{R} \texttt{SC.Kg}_\texttt{r}(\lambda_{sc})$
  $(m_0, m_1, \omega) \xleftarrow{R} \mathcal{A}_1^{\mathcal{O}}(\lambda_{sc}, pk_S^*, pk_R^*)$
  $C^* \leftarrow \texttt{SC.SignCrypt}(\lambda_{sc}, sk_S^*, pk_R^*, m_b)$
  $b' \xleftarrow{R} \mathcal{A}_2^{\mathcal{O}}(C^*, \omega)$
  Output $b'$
```

Fig. 8. Notions of confidentiality for (a) high-entropy signcryption schemes and (b) low-entropy signcryption schemes. Note that $\mathcal{A}_1$ may pass the state information ω to $\mathcal{A}_2$ in the lSCR game. The attacker's have access to a signcryption oracle $\texttt{SC.SignCrypt}(sk_S^*, \cdot, \cdot)$ and an unsigncryption oracle $\texttt{SC.UnSignCrypt}(\cdot, sk_R^*, \cdot)$.

The proof essentially shows that, since the challenge messages produced by a high-entropy attacker $\mathcal{A}_1$ have min-entropy $\mu(k)$, the probability that $\mathcal{A}_2$ queries the signcryption oracle on one of those messages is bounded by $4 \cdot q_s(k) \cdot \ell(k) \cdot \pi(k)$. If this does not occur, then a low-entropy attacker can easily run a high-entropy attacker as a black-box subroutine. The proof holds for deterministic schemes only. We are not aware if the same is true for probabilistic schemes.

We also have that the low-entropy confidentiality definition is strictly stronger than the high-entropy confidentiality definition. If SC is a high-entropy confidential signcryption scheme, then the signcryption scheme SC' given in Figure 9 is high-entropy confidential signcryption scheme but not a low-entropy confidential signcryption scheme.

```
$\texttt{SC.SignCrypt}'(sk_S, pk_R, m)$:
  $C \leftarrow \texttt{SC.SignCrypt}(sk_S, pk_R, m)$
  If $m = 0^k$
      Return $C\|0$
  Else
      Return $C\|1$
```

```
$\texttt{SC.UnSignCrypt}'(pk_S, sk_R, C)$:
  Parse $C$ as $C'\|c$ for $c \in \{0,1\}$
  $m \leftarrow \texttt{SC.UnSignCrypt}(pk_S, sk_R, C')$
  If $c = 0$ and $m \neq 0^k$
      Return $\perp$
  If $c = 1$ and $m = 0^k$
      Return $\perp$
  Else
      Return $m$
```

Fig. 9. A signcryption scheme which is high-entropy secure but not low-entropy secure

4.2 The Encrypt-and-Sign Signcryption Scheme

Initially, it may be thought that high-entropy confidentiality may be easily achieved through the combination of deterministic encryption and confidential

```
$\texttt{SC.Setup}(1^k)$
    $\lambda_{ss} \leftarrow \texttt{SS.Setup}(1^k)$
    $\lambda_{pke} \leftarrow \texttt{PKE.Setup}(1^k)$
    $\lambda_{sc} \leftarrow (\lambda_{ss}, \lambda_{pke})$
    Return $(\lambda_{sc})$

$\texttt{SC.Kg}_{\texttt{r}}(\lambda_{sc})$
    Parse $\lambda_{sc}$ as $(\lambda_{ss}, \lambda_{pke})$
    $(pk_R, sk_R) \leftarrow \texttt{PKE.Kg}(\lambda_{pke})$
    Return $(pk_R, sk_R)$

$\texttt{SC.Kg}_{\texttt{s}}(\lambda_{sc})$
    Parse $\lambda_{sc}$ as $(\lambda_{ss}, \lambda_{pke})$
    $(pk_S, sk_S) \leftarrow \texttt{SS.Kg}(\lambda_{ss})$
    Return $(pk_S, sk_S)$

$\texttt{SC.SignCrypt}(\lambda_{sc}, pk_R, sk_S, m)$
    Parse $\lambda_{sc}$ as $(\lambda_{ss}, \lambda_{pke})$
    $c \leftarrow \texttt{PKE.Enc}(\lambda_{pke}, pk_R, (pk_S||m))$
    $\sigma \leftarrow \texttt{SS.Sign}(\lambda_{ss}, sk_S, (pk_R||m))$
    Return $C = (c, \sigma)$

$\texttt{SC.UnSignCrypt}(\lambda_{sc}, sk_R, pk_S, C)$
    Parse $\lambda_{sc}$ as $(\lambda_{ss}, \lambda_{pke})$
    Parse $C$ as $(c, \sigma)$
    $(pk'_S||m') \leftarrow \texttt{PKE.Dec}(\lambda_{pke}, sk_R, c)$
    If $pk'_S \neq pk_S$, reject
    Extract $pk_R$ from $sk_R$
    If $\texttt{SS.Ver}(\lambda_{ss}, pk_S, (pk_R||m'), \sigma) = \perp$, reject
    Return $m'$
```

Fig. 10. The Encrypt-and-Sign signcryption scheme.

signatures. However, many of the classic composition theorems, such as encrypt-then-sign, fail to achieve high-entropy security even when instantiated with secure components.

However, we can show that the encrypt-and-sign (which is typically insecure as a signcryption scheme) is secure when instantiated with an IND-CCA2 public-key encryption scheme and a strongly confidential signature scheme[4]. The construction is given in Figure 10. The scheme can easily be shown to be unforgeable (in the sense that an attacker cannot obtain a signcryption of any message which was not previously sent by that sender to that receiver).

Theorem 1. *If the signature scheme is deterministic, strongly unforgeable, and strongly confidential, and the encryption scheme is* IND-CCA2 *secure, then the signcryption scheme is confidential in the high-entropy model. In particular, if there exists an attacker $\mathcal{A}$ against the high-entropy security of the signcryption scheme (asking $\ell(k)$ challenge messages and making at most $q_{sc}(k)$ signcryption queries), then there exist attackers $\mathcal{A}_{pke}$, $\mathcal{A}_{ss}$, and $\mathcal{A}_{sunf}$ against the* IND-CCA2 *security of the encryption scheme, against the strong confidentiality of the signature scheme, and against the strong unforgeability of the signature scheme, such that*

$$\mathsf{Adv}^{\mathrm{hSCR}}_{\mathsf{E+S},\mathcal{A}}(k) \leq \ell(k) \cdot \mathsf{Adv}^{\mathrm{cca2}}_{\mathsf{PKE},\mathcal{A}_{pke}}(k) + \mathsf{Adv}^{\mathrm{sSig}}_{\mathsf{SS},\mathcal{A}_{ss}}(k) + \mathsf{Adv}^{\mathrm{seuf-cma}}_{\mathsf{SS},\mathcal{A}_{sunf}}(k) \ .$$

where the running times of $\mathcal{A}_{pke}$, $\mathcal{A}_{ss}$, and $\mathcal{A}_{sunf}$ equal the one of $\mathcal{A}$ plus $(q_{sc}(k) + \ell(k)) \cdot (Time_{\texttt{SC.SignCrypt}}(k) + Time_{\texttt{SC.UnSignCrypt}}(k)) + O(k)$.

The security of this scheme can be proven in a manner similar to the encryption/signature composition theorems proven by An *et al.* [1].

[4] Strongly confidential, probabilistic signature schemes are given in Sections 3.3 and 3.4. These can be transformed in a strongly confidential, deterministic signature schemes using the derandomization techniques discussed in the next section.

4.3 Derandomization

Goldreich [17] presents a technique to turn any probabilistic signature scheme into a deterministic one. The idea is to include the secret key κ of a pseudorandom function ($\texttt{PRF.Kg}, \texttt{PRF}$) in the secret signing key and, when signing a message m, use the random coins $r = \texttt{PRF}(\kappa; m)$ in this process. Note that the resulting scheme now yields the same signature if run twice on the same message. A formal definition of a PRF can be found in Appendix A.

We show that Goldreich's idea applies to signcryption schemes as well, taking advantage of the fact that a signcryption scheme involves a secret signing key in which we can put the key κ of the pseudorandom function. Nonetheless, whereas a probabilistic signcryption scheme usually hides the fact that the same message has been encrypted twice, a derandomized version clearly leaks this information.

For a signcryption scheme SC the derandomized version $\mathsf{SC}^{\mathsf{PRF}}$ based on a pseudorandom function PRF works according to Goldreich's strategy:

```
$\texttt{SC.Setup}^{\mathsf{PRF}}(1^k)$:
  Return $\lambda_{sc} \leftarrow \texttt{SC.Setup}(1^k)$

$\texttt{SC.Kg}_{\texttt{s}}{}^{\mathsf{PRF}}(\lambda_{sc})$:
  $(sk_S, pk_S) \leftarrow \texttt{SC.Kg}_{\texttt{s}}(\lambda_{sc})$
  $\kappa \leftarrow \texttt{PRF.Kg}(1^k)$
  $sk_S^{\mathsf{PRF}} \leftarrow (sk_S, \kappa)$; $pk_S^{\mathsf{PRF}} \leftarrow pk_S$
  Return $(sk_S^{\mathsf{PRF}}, pk_S^{\mathsf{PRF}})$

$\texttt{SC.Kg}_{\texttt{r}}{}^{\mathsf{PRF}}(\lambda_{sc})$:
  Return $(sk_R, pk_R) \leftarrow \texttt{SC.Kg}_{\texttt{r}}$

$\texttt{SC.SignCrypt}^{\mathsf{PRF}}(sk_S^{\mathsf{PRF}}, pk_R, m)$:
  Parse $sk_S^{\mathsf{PRF}}$ as $(sk_S, \kappa)$
  $r \leftarrow \texttt{PRF}(\kappa, (pk_R, m))$
  $C \leftarrow \texttt{SC.SignCrypt}(sk_S, pk_R, m; r)$
      (i.e. using randomness $r$)
  Return $C$

$\texttt{SC.UnSignCrypt}^{\mathsf{PRF}}(sk_R, pk_S^{\mathsf{PRF}}, C)$:
  Return $\texttt{SC.UnSignCrypt}(sk_R, pk_S, C)$
```

Proposition 7 (Derandomized Signcryption). *Let* SC *be an unforgeable and high-entropy (resp. low-entropy) confidential signcryption scheme. Then the scheme* $\mathsf{SC}^{\mathsf{PRF}}$ *is a deterministic, unforgeable signcryption scheme which is high-entropy (resp. low-entropy) confidential. That is, for* $x \in \{l, h\}$ *and any adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against xSCR confidentiality, there exist adversaries* $\mathcal{D}$ *and* $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ *such that*

$$\mathsf{Adv}^{\mathrm{xSCR}}_{\mathsf{SC}^{\mathsf{PRF}},\mathcal{A}}(k) \leq 2 \cdot Adv^{\mathsf{PRF}}_{\mathcal{D}}(k) + \mathsf{Adv}^{\mathrm{xSCR}}_{\mathsf{SC},\mathcal{B}}(k) + 2q_{sc}(k) \cdot \ell(k) \cdot \pi(k)$$

where $\mathcal{D}$*'s running time is identical to the time of* $\mathcal{A}$*, plus* $Time_{\texttt{SC.Setup}}(k) + Time_{\texttt{SC.Kg}_{\texttt{s}}}(k) + Time_{\texttt{SC.Kg}_{\texttt{r}}}(k) + (q_{sc} + \ell(k)) \cdot Time_{\texttt{SC.SignCrypt}}(k) + O(k)$*; the running time of* $\mathcal{B}$ *equals the time of* $\mathcal{A}$ *plus* $O(q_{sc} \cdot \log q_{sc})$*.*

Acknowledgements. The authors wish to thank the ECRYPT II MAYA working group on the design and analysis of primitives and protocols for interesting preliminary discussions on this topic. The work described in this report has in part been supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 ECRYPT II. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. Dominique and Marc were supported by the Emmy Noether grant Fi 940/2-1 of the German Research Foundation (DFG), and by CASED (`www.cased.de`).

References

1. 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.
2. J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. *Journal of Cryptology*, 20(2):203–235, 2007.
3. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in Cryptology – Crypto 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer-Verlag, 2007.
4. M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In D. Wagner, editor, *Advances in Cryptology – Crypto 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 360–378. Springer-Verlag, 2008.
5. M. Bellare and P. Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology – Eurocrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.
6. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *Advances in Cryptology – Crypto 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 335–359. Springer-Verlag, 2008.
7. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. Kaliski, editor, *Advances in Cryptology – Crypto '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer-Verlag, 1997.
8. R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions. In *Proc. 30th Symposium on the Theory of Computing – STOC 1998*, pages 131–140. ACM, 1998.
9. J.-S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *Advances in Cryptology – Crypto 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer-Verlag, 2000.
10. A. W. Dent, M. Fischlin, M. Manulis, M. Stam, and D. Schröder. Confidential signatures and deterministic signcryption. Available from `http://eprint.iacr.org/2009/588`, 2009.
11. Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin. Randomness extraction and key derivation using the CBC, Cascade and HMAC modes. In M. Franklin, editor, *Advances in Cryptology – Crypto 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 494–510. Springer-Verlag, 2004.
12. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In J. Kilian, editor, *Theory of Cryptography – TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 556–577. Springer-Verlag, 2005.
13. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

14. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *Advances in Cryptology – Crypto '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1986.
15. M. Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In J. Stern, editor, *Advances in Cryptology - Eurocrypt 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 429–444. Springer-Verlag, 1999.
16. Marc Fischlin. Anonymous signatures made easy. In *Public-Key Cryptography (PKC) 2007*, volume 4450 of *Lecture Notes in Computer Science*, pages 31–42. Springer-Verlag, 2007.
17. O. Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In A. M. Odlyzko, editor, *Proceedings on Advances in Cryptology – Crypto '86*, volume 263 of *Lecture Notes in Computer Science*, pages 104–110. Springer-Verlag, 1987.
18. N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, 2001.
19. N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Science*, 58(1):148–173, 1999.
20. N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Science*, 52(1):43–52, 1996.
21. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology – Crypto '91*, volume 576 of *Lecture Notes in Computer Science*, pages 434–444. Springer-Verlag, 1991.
22. A. Russell and H. Wang. How to fool an unbounded adversary with a short key. In L. Knudsen, editor, *Advances in Cryptology – Eurocrypt 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 133–148. Springer-Verlag, 2002.
23. G. Yang, D. S. Wong, X. Deng, and H. Wang. Anonymous signature schemes. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 347–363. Springer-Verlag, 2006.
24. Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ 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.

A Standard Security Notions

A.1 Signature Schemes

The standard notion for signature security is that of (strong) existential unforgeability under chosen message attacks (sEUF-CMA). The strong version is defined below. Freshness of (m, σ) indicates that σ was never received by $\mathcal{A}$ as response to a signing request on m.

$$\mathsf{Adv}^{\mathrm{seuf-cma}}_{\mathsf{SS},\mathcal{A}}(k) = \Pr\left[\begin{array}{l} \mathtt{SS.Ver}(\lambda_{ss}, pk, m, \sigma) = \top \\ (m, \sigma) \text{ is fresh} \end{array} : \begin{array}{l} \lambda_{ss} \xleftarrow{R} \mathtt{SS.Setup}(1^k) \\ (pk, sk) \xleftarrow{R} \mathtt{SS.Kg}(\lambda_{ss}) \\ (m, \sigma) \xleftarrow{R} \mathcal{A}^{\mathtt{SS.Sign}(\lambda_{ss}, sk, \cdot)}(\lambda_{pke}, pk) \end{array}\right].$$

The advantage $\mathsf{Adv}^{\mathrm{euf-cma}}_{\mathsf{SS},\mathcal{A}}(k)$ of the slightly weaker notion (EUF-CMA) is defined analogously, but this time m only needs to be fresh.

A.2 Public-Key Encryption

A *public key encryption* scheme is a tuple of algorithms $\mathsf{PKE} = (\texttt{PKE.Setup}, \texttt{PKE.Kg}, \texttt{PKE.Enc}, \texttt{PKE.Dec})$. First the common parameters for the given security level $k \in \mathbb{N}$ are generated by $\lambda_{pke} \xleftarrow{R} \texttt{PKE.Setup}(1^k)$ after which a user's public/private keys are generated using $(pk, sk) \xleftarrow{R} \texttt{PKE.Kg}(\lambda_{pke})$. Given such a key pair, a message $m \in \{0,1\}^*$ is encrypted by $c \xleftarrow{R} \texttt{PKE.Enc}(\lambda_{pke}, pk, m)$; a ciphertext is decrypted by $m \xleftarrow{R} \texttt{PKE.Dec}(\lambda_{pke}, sk, c)$. For consistency, we require that for all messages $m \in \{0,1\}^*$, we have that $\texttt{PKE.Dec}(sk, \texttt{PKE.Enc}(pk, m)) = m$.

We require a PKE is secure against IND-CCA2 attacks [21, 13], for which the advantage of an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is defined as

$$\mathsf{Adv}^{\mathrm{cca2}}_{\mathsf{PKE},\mathcal{A}}(k) = \left|\Pr\left[\, \mathit{Expt}^{\mathit{cca2-0}}_{\mathcal{A}} = 1\right] - \Pr\left[\, \mathit{Expt}^{\mathit{cca-1}}_{\mathcal{A}} = 1\right]\right| \ ,$$

where (for $b \in \{0,1\}$):

```
$\mathit{Expt}^{\mathit{cca2-b}}_{\mathcal{A}}$
  $\lambda_{pke} \xleftarrow{R} \texttt{PKE.Setup}(1^k)$
  $(pk, sk) \xleftarrow{R} \texttt{PKE.Kg}(\lambda_{pke})$
  $(m_0, m_1, \omega) \xleftarrow{R} \mathcal{A}_1^{\texttt{PKE.Dec}(\lambda_{pke}, sk, \cdot)}(\lambda_{pke}, pk)$
  $c^* \xleftarrow{R} \texttt{PKE.Enc}(\lambda_{pke}, pk, m_b)$
  $b' \xleftarrow{R} \mathcal{A}_2^{\texttt{PKE.Dec}(\lambda_{pke}, sk, \cdot)}(c^*, \omega)$
  Output 1 if $b' = b$
```

The adversary $\mathcal{A}_2$ is may not query $\texttt{PKE.Dec}(sk, \cdot)$ with c^*. A PKE scheme PKE is IND-CCA2 secure if the advantage function $\mathsf{Adv}^{\mathrm{cca2}}_{\mathsf{PKE},\mathcal{A}}(k)$ is a negligible function for all probabilistic polynomial-time adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

A.3 Pseudo-Random Functions

A pseudo-random function is a pair of algorithms $\mathsf{PRF} = (\texttt{PRF.Kg}, \texttt{PRF})$. The key generation algorithm outputs a key $\kappa \xleftarrow{R} \texttt{PRF.Kg}(1^k)$. For our purposes, a pseudo-random function $\texttt{PRF}(\kappa, \cdot)$ takes arbitrary bitstrings as inputs and outputs a bitstring in a given space $\mathcal{R}$. Let $\mathcal{F}$ be the set of all functions from $f : \{0,1\}^* \to \mathcal{R}$. The security of a PRF against a PPT attacker $\mathcal{A}$ is defined by the following two games:

```
$\mathit{Expt}^{\mathit{PRF-0}}_{\mathcal{A}}(k)$:
  $\kappa \xleftarrow{R} \texttt{PRF.Kg}(1^k)$
  Return $\mathcal{A}^{\texttt{PRF}(\kappa,\cdot)}(1^k)$

$\mathit{Expt}^{\mathit{PRF-1}}_{\mathcal{A}}(k)$:
  $f \xleftarrow{R} \mathcal{F}$
  Return $\mathcal{A}^{f(\cdot)}(1^k)$
```

The attacker's advantage is defined to be:

$$\mathit{Adv}^{\mathsf{PRF}}_{\mathsf{PRF},\mathcal{A}}(k) = |\Pr[\mathit{Expt}^{\mathit{PRF-0}}_{\mathcal{A}}(k) = 1] - \Pr[\mathit{Expt}^{\mathit{PRF-1}}_{\mathcal{A}}(k) = 1]| \, .$$