Secure Cryptographic Workflow in the Standard Model

M. Barbosa[1] and P. Farshim[2]

[1] Departamento de Informática, Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal.
mbb@di.uminho.pt

[2] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
farshim@cs.bris.ac.uk

Abstract. Following the work of Al-Riyami et al. we define the notion of key encapsulation mechanism supporting cryptographic workflow (WF-KEM) and prove a KEM-DEM composition theorem which extends the notion of hybrid encryption to cryptographic workflow. We then generically construct a WF-KEM from an identity-based encryption (IBE) scheme and a secret sharing scheme. Chosen ciphertext security is achieved using one-time signatures. Adding a public-key encryption scheme we are able to modify the construction to obtain escrow-freeness. We prove all our constructions secure in the standard model.

Keywords. Cryptographic Workflow. Key Encapsulation. Secret Sharing. Identity-Based Encryption.

1 Introduction

The term *workflow* is used to describe a system in which actions must be performed in a particular order. In *cryptographic workflow* [23] this is achieved by making decryption a privileged action which can only be executed by users which possess an appropriate set of *authorisation credentials*, or simply *credentials*. Credentials are issued by a set of *authorisation authorities*, which can ensure that some action has been performed, or that some event has occurred, before granting them to users. Restricting access to encrypted messages in this way, workflow mechanisms can be implemented with cryptographic security guarantees.

An encryption scheme supporting cryptographic workflow should provide the following functionality [1]. Alice specifies the credentials that Bob should have in a *policy* that she decides before encrypting. Alice should be able to perform this encryption without knowing what credentials Bob actually has. A particular authorisation authority will validate that Bob is entitled to a given credential before awarding it. Each credential acts as a (partial) decryption key. Alice may also want to be sure that no colluding set of these authorisation authorities is able to decrypt and recover the message that she intended for Bob. If this is the case, the system should be *escrow-free*.

In this paper we introduce the notion of KEMs supporting cryptographic workflow (WF-KEM) and their escrow-free counterparts (EFWF-KEM). We adapt the security models proposed in [1] for encryption schemes accordingly. We argue that the KEM-DEM paradigm introduced by Cramer and Shoup [14] for public-key encryption schemes also applies when one moves to encryption schemes supporting cryptographic workflow. In fact, we show that combining a secure WF-KEM (EFWF-KEM) with a secure DEM, one obtains a secure (escrow-free) encryption scheme supporting cryptographic workflow.

We present a generic construction that permits building WF-KEMs out of simpler cryptographic primitives. This is a generalisation of the construction presented in [1] based on the identity-based encryption (IBE) scheme of Boneh and Franklin. We show how one can construct analogous schemes by replacing its building blocks with other components providing the same functionality. More specifically, we prove that our transformation permits constructing a secure WF-KEM using secure IBE and Secret Sharing (SS) schemes. Finally, we extend our generic construction to obtain an EFWF-KEM using a secure public-key encryption scheme. Chosen ciphertext security is achieved via a one-time signature scheme. Our constructions are all secure in the standard model.

The paper is structured as follows. We first review related work in Section 2 and present the cryptographic primitives we use as building blocks in Section 3. Then in Section 4 we define precisely what we mean by secure WF-KEMs and EFWF-KEMs. In Section 5 we propose generic constructions of these primitives and prove them secure. Finally, in Section 6, we analyse the implications and efficiency of our results for cryptographic workflow and related problems.

2 Related work

Identity-based cryptography was initially proposed by Shamir [26], who also introduced the first identity-based signature scheme. The first practical identity-based encryption (IBE) scheme is that proposed by Boneh and Franklin in [7], whose operation relies on the use of bilinear maps over groups of points on an elliptic curve. Sakai and Kasahara [24] later proposed another IBE scheme, also based on bilinear maps, but adopting a different key construction. The security of this scheme was established by Chen *et al.* in [11]. The latter scheme allows for more efficient encryption operation. Both these schemes are secure in the random oracle model (ROM). Recently, Waters [28], Kiltz [20] and Gentry [17] have proposed practical IBE schemes which are secure in the standard model.

The KEM-DEM construction was formalised by Cramer and Shoup in [14]. It captures the concept of hybrid encryption whereby one constructs a public-key encryption scheme by combining a symmetric Data Encapsulation Mechanism (DEM) with an asymmetric Key Encapsulation Mechanism (KEM). The security of the hybrid construction depends, of course, on the security of the KEM and DEM. In [14] it is shown that if the KEM and DEM constructions are individually secure, the resulting public-key encryption scheme will be also secure. The relations between the security notions for KEMs and the conditions for the security of KEM/DEM constructions are further discussed in [22, 18] respectively. Dent [15] describes several constructions for secure KEMs. The KEM-DEM paradigm has been extended to the identity-based setting in [6].

Cryptographic workflow follows from the original ideas by Chen et al. in [12, 13]. There the authors explored the possibilities of using the Boneh and Franklin IBE scheme in a setting where a user can extract different identity-based private keys from multiple TAs. They proposed using *credential descriptors* as public keys, in place of the usual identity strings, and showed that combining the master public keys of the TAs in different ways, one may securely send a message to a recipient and restrict her ability to decrypt it with a high degree of flexibility. Smart [27] applied the same principle to access control. Paterson [23] first employed the term *workflow* to describe this type of scheme.

Key escrow is an inherent property of identity-based cryptography, since it is the TA that computes private keys. This may be a problem in some applications. To solve this (and the issue of certificate management), Al-Riyami and Paterson [2] propose *certificateless public-key cryptography* (CL-PKC). CL-PKC is a modification of identity-based techniques which requires each user to have a (possibly unauthenticated) public key. Messages are encrypted using a combination of a user's public key and its identity.

Al-Riyami et al. [1] formalised the definitions of primitives and security models associated with cryptographic workflow and proposed an efficient escrow-free encryption scheme supporting cryptographic workflow. The scheme is based on the Boneh and Franklin IBE and it is proved secure under two security notions. The first one, called *receiver security*, ensures that only users with an appropriate set of credentials can decrypt the message. The second, called *external security*, captures the escrow-freeness notion: it must be unfeasible for any colluding set of TAs to decrypt the message. Unlike CL-PKC, however, escrow-freeness is achieved using a classical public-key encryption layer which relies on public key certification to achieve security.

Encryption schemes supporting cryptographic workflow are very close to those associated with *hidden credential* systems [9, 19]. Both types of schemes typically employ a secret sharing layer and an identity-based encryption layer, although the goals in each case are different. In hidden credential systems one seeks to keep the access control policy secret, whereas in workflow schemes this is not the case. Secret sharing schemes are covered in [5, 21, 25].

A common feature of many schemes proposed for CL-PKC, cryptographic workflow and hidden credentials is that they are based on the concept of multiple encryption (or re-encryption). In multiple encryption, a ciphertext is created by combining the results of several instances of an encryption algorithm with different encryption keys. In the simplest case, where only two decryption keys are involved, the objective is that even if the adversary is in possession of one of those keys, she obtains no advantage. Recently, Dodis and Katz [16] have addressed the chosen ciphertext security of multiple encryptions in the general case, and have proposed generic constructions which are semantically secure. Our constructions build on these results.

3 Building Blocks

3.1 Public-Key Encryption

A public-key encryption (PKE) scheme [14] is specified by three PT algorithms:

- $\mathbb{G}_{\mathtt{PKE}}(1^\kappa)$: A PPT algorithm which takes as input 1^κ and returns a secret key $\mathtt{SK}$ and a public key $\mathtt{PK}$ as well as the descriptions of the message, randomness and ciphertext spaces[3].
- $\mathbb{E}_{\mathtt{PKE}}(\mathtt{m}, \mathtt{PK})$: This is the PPT encryption algorithm, which on input of a message $\mathtt{m} \in \mathbb{M}_{\mathtt{PKE}}(\mathtt{PK})$ and a public key $\mathtt{PK}$, outputs a ciphertext $\mathtt{c}$.
- $\mathbb{D}_{\mathtt{PKE}}(\mathtt{c}, \mathtt{SK})$: This is the deterministic decryption algorithm. On input of a ciphertext $\mathtt{c}$ and a private key $\mathtt{SK}$ this outputs a message $\mathtt{m} \in \mathbb{M}_{\mathtt{PKE}}(\mathtt{PK})$ or a failure symbol $\perp$.

Informally, the soundness of a PKE scheme requires that the decryption algorithm recovers the correct plaintext with overwhelming probability, when provided with a valid ciphertext and the correct decryption key.

The semantic security of a public-key scheme against adaptive chosen ciphertext attacks is defined through the following indistinguishability game.

```
IND-CCA2
 1. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}_{\mathtt{PKE}}(1^\kappa)$
 2. $(s, \mathtt{m}_0, \mathtt{m}_1) \leftarrow A_1^{\mathcal{O}_1}(\mathtt{PK})$
 3. $b \leftarrow \{0, 1\}$
 4. $\mathtt{c}^* \leftarrow \mathbb{E}_{\mathtt{PKE}}(\mathtt{m}_b, \mathtt{PK})$
 5. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathtt{c}^*, s)$
```

$$\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-CCA2}}(A) := |\Pr[b' = b] - 1/2|.$$

Here, $\mathcal{O}_1$ and $\mathcal{O}_2$ denote a decryption oracle with the restriction that, $\mathcal{O}_2$ cannot be queried on $\mathtt{c}^*$.

A PKE scheme is called IND-CCA2 secure if all PPT attackers have negligible advantage as a function of the security parameter[4].

3.2 Data Encapsulation Mechanism

A data encapsulation mechanism (DEM) is a one-time secret-key encryption (SKE) scheme, where the symmetric key is used to encrypt a single message. More formally a DEM/SKE is specified by three PT algorithms:

- $\mathbb{G}_{\mathtt{DEM}}(1^\kappa)$: This is the probabilistic key generation algorithm which on input of a security parameter 1^κ outputs a key $\mathtt{k} \in \mathbb{K}_{\mathtt{DEM}}(1^\kappa)$.

[3] These are denoted by $\mathbb{M}_{\mathtt{PKE}}(\mathtt{PK})$, $\mathbb{R}_{\mathtt{PKE}}(\mathtt{PK})$ and $\mathbb{C}_{\mathtt{PKE}}(\mathtt{PK})$ respectively. From now on we assume that the (master) public key of various primitives in this paper includes these as well as the security parameter.

[4] This will be the general definition of security in the rest of this paper, once advantage is defined.

- $\mathbb{E}_{\mathtt{DEM}}(\mathtt{m},\mathtt{k})$: This is the probabilistic encapsulation algorithm which on input of a message $\mathtt{m} \in \{0,1\}^*$ and a key $\mathtt{k} \in \mathbb{K}_{\mathtt{DEM}}(1^\kappa)$, outputs a ciphertext $\mathtt{c}$.
- $\mathbb{D}_{\mathtt{DEM}}(\mathtt{c},\mathtt{k})$: This is the deterministic decryption algorithm which on input of a ciphertext $\mathtt{c}$ and a key $\mathtt{k} \in \mathbb{K}_{\mathtt{DEM}}(1^\kappa)$ outputs a message $\mathtt{m} \in \{0,1\}^*$ or a failure symbol $\perp$.

Such a scheme is called sound if $\mathbb{D}_{\mathtt{DEM}}(\mathbb{E}_{\mathtt{DEM}}(\mathtt{m},\mathtt{k}),\mathtt{k}) = \mathtt{m}$. The Find-then-Guess security of a DEM is defined through a game similar to the IND-CCA2 game for a PKE scheme, with the difference that only a second stage attack is permitted:

```
FG-CCA
 1. $(s, \mathtt{m}_0, \mathtt{m}_1) \leftarrow A_1(1^\kappa)$
 2. $\mathtt{k} \leftarrow \mathbb{G}_{\mathtt{DEM}}(1^\kappa)$
 3. $b \leftarrow \{0,1\}$
 4. $\mathtt{c}^* \leftarrow \mathbb{E}_{\mathtt{DEM}}(\mathtt{m}_b, \mathtt{k})$
 5. $b' \leftarrow A_2^{\mathcal{O}_D}(\mathtt{c}^*, s)$
```

$$\mathrm{Adv}_{\mathtt{DEM}}^{\mathtt{FG-CCA}}(A) := |\Pr[b' = b] - 1/2|.$$

In the above we require that $\mathtt{m}_0$ and $\mathtt{m}_1$ are of the same length. The oracle $\mathcal{O}_D$ denotes a decapsulation oracle subject to the rule that it cannot be queried on $\mathtt{c}^*$. We only need this weak definition of CCA security as the key used to encrypt is randomly chosen after the message.

A secure DEM can be constructed using a one-time pad, where the key is expanded using a pseudo-random generator and a one-time MAC is used to provide message authenticity [14].

3.3 Identity-Based Encryption

An identity-based encryption (IBE) scheme [7] is specified by four polynomial time algorithms:

- $\mathbb{G}_{\mathtt{IBE}}(1^\kappa)$: A PPT algorithm which takes as input 1^κ and returns the TA's master secret key $\mathtt{Msk}$ and a matching master public key $\mathtt{Mpk}$. This algorithm also outputs descriptions of the message, ciphertext and randomness spaces of an IBE scheme, denoted by $\mathbb{M}_{\mathtt{IBE}}(\mathtt{Mpk})$, $\mathbb{C}_{\mathtt{IBE}}(\mathtt{Mpk})$ and $\mathbb{R}_{\mathtt{IBE}}(\mathtt{Mpk})$ respectively. These are parameterised by the master public key $\mathtt{Mpk}$ and implicitly by the security parameter κ.
- $\mathbb{X}_{\mathtt{IBE}}(\mathtt{ID},\mathtt{Msk})$: The PPT private key extraction algorithm which takes as input $\mathtt{Msk}$ and $\mathtt{ID} \in \{0,1\}^*$, an identifier string for a user, and returns the associated private key $S_{\mathtt{ID}}$.
- $\mathbb{E}_{\mathtt{IBE}}(\mathtt{m},\mathtt{ID},\mathtt{Mpk})$: This is the PPT encryption algorithm. On input of a message $\mathtt{m} \in \mathbb{M}_{\mathtt{IBE}}(\mathtt{Mpk})$, an identifier $\mathtt{ID}$ and the master public key $\mathtt{Mpk}$, this algorithm outputs $\mathtt{c} \in \mathbb{C}_{\mathtt{IBE}}(\mathtt{Mpk})$.
- $\mathbb{D}_{\mathtt{IBE}}(\mathtt{c}, S_{\mathtt{ID}})$: This is the deterministic decryption algorithm. On input of a ciphertext c and a private key $S_{\mathtt{ID}}$ this outputs a message $\mathtt{m} \in \mathbb{M}_{\mathtt{IBE}}(\mathtt{Mpk})$ or a failure symbol $\perp$.

An IBE scheme is called *sound* if for all messages and user identities in the appropriate message and identity spaces we have:

$$\Pr\left(\mathtt{m} = \mathbb{D}_{\mathtt{IBE}}(c, S_{\mathtt{ID}}) \middle| \begin{array}{l} (\mathtt{Msk}, \mathtt{Mpk}) \leftarrow \mathbb{G}_{\mathtt{IBE}}(1^\kappa) \\ c \leftarrow \mathbb{E}_{\mathtt{IBE}}(\mathtt{m}, \mathtt{ID}, \mathtt{Mpk}) \\ S_{\mathtt{ID}} \leftarrow \mathbb{X}_{\mathtt{IBE}}(\mathtt{ID}, \mathtt{Msk}) \end{array}\right) = 1.$$

The indistinguishability game in an attack model $\mathtt{atk}$ for an IBE scheme is:

```
IND-atk
 1. $(\mathtt{Msk}, \mathtt{Mpk}) \leftarrow \mathbb{G}_{\mathtt{IBE}}(1^\kappa)$
 2. $(s, \mathtt{m}_0, \mathtt{m}_1, \mathtt{ID}^*) \leftarrow A_1^{\mathcal{O}_1}(\mathtt{Mpk})$
 3. $b \leftarrow \{0, 1\}$
 4. $\mathtt{c}^* \leftarrow \mathbb{E}_{\mathtt{IBE}}(\mathtt{m}_b, \mathtt{ID}^*, \mathtt{Mpk})$
 5. $b' \leftarrow A_2^{\mathcal{O}_2}(c^*, s)$
```

$$\mathrm{Adv}_{\mathtt{IBE}}^{\mathtt{IND-atk}}(A) := |\Pr[b' = b] - 1/2|.$$

Here $\mathtt{atk} \in \{\mathrm{CPA}, \mathrm{CCA1}, \mathrm{CCA2}\}$ denotes the attack model. These are defined as usual by allowing the adversary to have access to various oracles in each stage: (1) in CPA model: $\mathcal{O}_1 = \mathcal{O}_2 = \mathcal{O}_X$; (2) in CCA1 model: $\mathcal{O}_1 = \{\mathcal{O}_X, \mathcal{O}_D\}, \mathcal{O}_2 = \mathcal{O}_X$; (3) in CCA2 model: $\mathcal{O}_1 = \mathcal{O}_2 = \{\mathcal{O}_X, \mathcal{O}_D\}$. Here $\mathcal{O}_X$ and $\mathcal{O}_D$ denote extraction and decryption oracles, subject to the rule that they cannot be queried on $\mathtt{ID}^*$ and $(\mathtt{c}^*, \mathtt{ID}^*)$ respectively.

3.4 Secret Sharing

We follow the approach in [5] for secret sharing over general access structures.

Definition 1. *A collection $\mathcal{P}$ of subsets of a set $P = \{X_1, \ldots, X_n\}$ is called a monotone access structure on P if:*

$$\forall A \in \mathcal{P} \textit{ and } \forall B \subseteq P, A \subseteq B \Rightarrow B \in \mathcal{P}.$$

A set $Q \subseteq P$ is called a qualifying subset of P if $Q \in \mathcal{P}$.

The access structures considered in this paper are all monotone and non-trivial i.e. $\mathcal{P} \neq \emptyset$. Note that non-triviality implies $P \in \mathcal{P}$.

A secret sharing scheme is defined as a pair of algorithms as follows:

- $\mathbb{S}(1^\kappa, \mathtt{s}, \mathcal{P})$: This is the probabilistic secret sharing algorithm which on input of the security parameter 1^κ, a string $\mathtt{s}$ and a (monotone) access structure $\mathcal{P}$, outputs a list of shares $\mathbf{shr} = (\mathtt{shr}_1, \ldots, \mathtt{shr}_n)$ one for each element in $P = \{X_1, \ldots, X_n\}$ as well as some auxiliary information $\mathtt{aux}$.
- $\mathbb{S}^{-1}(\mathbf{shr}, \mathtt{aux})$: This is the deterministic secret reconstruction algorithm. On input of a list of shares $\mathbf{shr}$ and some auxiliary information $\mathtt{aux}$, outputs a secret $\mathtt{s}$ or a failure symbol $\perp$.

A secret sharing scheme is *sound* if for all access structures $\mathcal{P}$ and strings $\mathtt{s} \in \{0,1\}^*$ of polynomial length in κ, we have:

$$\Pr\left(\mathtt{s} = \mathbb{S}^{-1}(\mathbf{shr}', \mathtt{aux}) \left| \begin{array}{l} (\mathbf{shr}, \mathtt{aux}) \leftarrow \mathbb{S}(1^\kappa, \mathtt{s}, \mathcal{P}) \\ Q \leftarrow \mathcal{P} \\ \text{Parse } (X_{i_1}, \dots, X_{i_k}) \leftarrow Q \\ [\mathbf{shr}']_j \leftarrow [\mathbf{shr}]_{i_j}, 1 \leq j \leq k \end{array} \right.\right) = 1.$$

The level of security provided by the secret sharing scheme will influence the overall security of our constructions. We consider both perfect and computational (non-perfect) secret sharing schemes [21].

For perfect secret sharing we will not require a game-based security definition. When necessary, we use an information theoretical argument based on the following definition of security.

Definition 2. *(Perfect Secret Sharing) A secret sharing scheme provides perfect secrecy if every non-qualifying subset of shares does not contain any information about the secret (in the information-theoretic sense). Formally, for any non-empty and non-qualifying set $\{i_1, \dots, i_n\}$ of an access structure $\mathcal{P}$, and for every two secrets* $\sec_0$ *and* $\sec_1$*, let* $(\mathtt{aux}_b, \mathbf{shr}_b) \leftarrow \mathbb{S}(\sec_b, \mathcal{P})$*, for* $b \in \{0,1\}$*. Then, for every possible share value* $\mathtt{shr}_{i_j}$*,* $1 \leq j \leq n$ *and for every possible* $\mathtt{aux}$ *value*

$$\Pr[\mathtt{shr}_{i_j} = [\mathbf{shr}_0]_{i_j}] = \Pr[\mathtt{shr}_{i_j} = [\mathbf{shr}_1]_{i_j}] \text{ and } \Pr[\mathtt{aux} = \mathtt{aux}_0] = \Pr[\mathtt{aux} = \mathtt{aux}_1].$$

Note that, for perfect secret sharing schemes we do not have an asymptotic definition of security, and therefore we drop the security parameter in the primitive definition.

In perfect secret sharing schemes, the secret size constitutes a lower bound on the individual size of shares. To reduce this lower bound, one must relax the security definition and settle for polynomial-time indistinguishability. For computational secret sharing, we shall use the following definitions of semantic security: secret indistinguishability against selective share attacks (IND-SSA), and against adaptive share attacks (IND-CSA).

IND-SSA	IND-CSA
1. $(s, \mathtt{s}_0, \mathtt{s}_1, \mathcal{P}^*, i_1, \dots, i_k) \leftarrow A_1(1^\kappa)$	1. $(s, \mathtt{s}_0, \mathtt{s}_1, \mathcal{P}^*) \leftarrow A_1(1^\kappa)$
2. $b \leftarrow \{0,1\}$	2. $b \leftarrow \{0,1\}$
3. $(\mathbf{shr}^*, \mathtt{aux}^*) \leftarrow \mathbb{S}(1^\kappa, \mathtt{s}_b, \mathcal{P}^*)$	3. $(\mathbf{shr}^*, \mathtt{aux}^*) \leftarrow \mathbb{S}(1^\kappa, \mathtt{s}_b, \mathcal{P}^*)$
4. $b' \leftarrow A_2(\mathtt{aux}^*, ([\mathbf{shr}^*]_{i_j})_{j=1}^k, s)$	4. $b' \leftarrow A_2^{\mathcal{O}}(\mathtt{aux}^*, s)$

$$\mathrm{Adv}_{\mathtt{SS}}^{\mathtt{IND-atk}}(A) := |\Pr[b' = b] - 1/2|.$$

Here $\mathtt{atk} \in \{\mathtt{SSA}, \mathtt{CSA}\}$. In the SSA model, $k \leq n$, and $\{i_1, \dots, i_k\}$ must not include a qualifying set of shares in $\mathcal{P}^*$. In the CSA model, $\mathcal{O}$ is a share extraction oracle subject to the condition that the adversary cannot extract a set of shares corresponding to a qualifying set in $\mathcal{P}^*$.

3.5 One-Time Signature

In our constructions we achieve chosen ciphertext security using an adaptation of the technique by Canetti et al. [10] which is based on a one-time signature (OTS) scheme. An OTS is a weak form of signature in which the signing/verification key pair can only be used once. More specifically, an OTS is defined by a three-tuple of PPT algorithms:

- $\mathbb{G}_{\mathtt{OTS}}(1^\kappa)$: This is the key generation algorithm which, on input of the security parameter, outputs a key pair $(\mathtt{vk}, \mathtt{sk})$.
- $\mathtt{Sig}(\mathtt{m}, \mathtt{sk})$: This is the signature algorithm, which takes a message $\mathtt{m}$ and a secret key $\mathtt{sk}$, and returns a signature σ.
- $\mathtt{Ver}(\mathtt{m}, \sigma, \mathtt{vk})$: This is the deterministic verification algorithm which, given a message, a signature σ and a verification key returns either 0 (*reject*) or 1 (*accept*).

The *strong unforgeability* security of an OTS is defined through the following game in which any PPT adversary must have negligible advantage.

```
UF
 1. $(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathbb{G}_{\mathtt{OTS}}(1^\kappa)$
 2. $(\mathtt{m}, s) \leftarrow A_1(\mathtt{vk})$
 3. $\sigma \leftarrow \mathtt{Sig}(\mathtt{sk}, \mathtt{m})$
 4. $(\mathtt{m}', \sigma') \leftarrow A_2(s, \sigma)$
```

$$\mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(A) := \Pr[(\sigma', \mathtt{m}') \neq (\mathtt{m}, \sigma) \wedge \mathtt{Ver}(\mathtt{m}', \sigma', \mathtt{vk}) = 1].$$

Note that this unforgeability definition implies that it must be unfeasible to create a new valid signature for a previously signed message. OTS schemes meeting this security definition can be constructed from any one-way function.

4 KEM Primitives for Cryptographic Workflow

4.1 Access Structures, Policies and Credentials

We first explain how we treat access structures in our constructions. We follow an approach similar to that in [1], but we briefly clarify this point stating our assumptions on their meaning in real life.

Suppose that we would like to encrypt a message such that only British nationals can read. To achieve this, we need a TA who issues *credentials* only to those who possess British nationality. For example, the Home Office would be the obvious TA to issue British Nationality certificates. However, it could be the case that two or more TAs are able to issue such a credential. For instance, the user's employer could, after checking the appropriate documentation, grant her a similar credential. We therefore need to specify precisely which authority we are trusting. The need for this is even more apparent when the policy is more complex. Consider the policy $\mathtt{English} \wedge \mathtt{English} \wedge \mathtt{Adult}$, where the first two terms refer to nationality and language with credentials issued by the Home Office and the British Council, respectively. It could also be the case that the same authority issues credentials on age and nationality: it is up to the authority to interpret the semantics.

For this reason, we view a policy term as a pair $(\mathtt{ID}, \mathtt{Mpk})$ where $\mathtt{ID} \in \{0,1\}^*$ is an identifier for the policy term and $\mathtt{Mpk}$ is the public key of the authority issuing the credential described in $\mathtt{ID}$. We denote by m the number of distinct TAs present in the system, by n the number of distinct policy terms in an access structure and by k the number of distinct policy terms in a qualifying set.

4.2 KEMs Supporting Cryptographic Workflow

A *key encapsulation mechanism supporting cryptographic workflow* (WF-KEM) is defined as a four-tuple of polynomial time (PT) algorithms as follows:

- $\mathbb{G}_{\mathtt{WF-KEM}}(1^\kappa, m)$: This is the probabilistic authority key generation algorithm which on input of a security parameter 1^κ outputs m authority secret/public key pairs $((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^m$, as well as the descriptions of the key, randomness and ciphertext spaces. These are denoted by $\mathbb{K}_{\mathtt{WF-KEM}}$, $\mathbb{R}_{\mathtt{WF-KEM}}$ and $\mathbb{C}_{\mathtt{WF-KEM}}$, respectively.
- $\mathbb{X}_{\mathtt{WF-KEM}}(X, \mathtt{Msk})$: This is the probabilistic credential extraction algorithm which on input of a policy term X, consisting of a policy identifier/authority public key pair $(\mathtt{ID}, \mathtt{Mpk})$, and the secret key $\mathtt{Msk}$ corresponding to $\mathtt{Mpk}$, outputs a pair $(\mathtt{crd}, X)$ which we call a *credential.*
- $\mathbb{E}_{\mathtt{WF-KEM}}(\mathcal{P})$: This is the probabilistic key encapsulation algorithm which on input of an access structure $\mathcal{P}$ on n policy terms $P = \{X_1, \dots, X_n\}$ outputs a pair $(\mathtt{k}, \mathbf{c})$ where $\mathtt{k} \in \mathbb{K}_{\mathtt{WF-KEM}}$ and $\mathbf{c}$ is an encapsulation of $\mathtt{k}$.
- $\mathbb{D}_{\mathtt{WF-KEM}}(\mathbf{c}, \mathbf{crd})$: This is the deterministic decapsulation algorithm which on input of an encapsulation $\mathbf{c}$ and a list of k credentials $\mathbf{crd}$, outputs a key or a failure symbol $\perp$.

A WF-KEM scheme is called *sound* if for every policy $\mathcal{P}$ on n terms with $m, n \in \mathbb{N}$ we have:

$$\Pr\left(\mathtt{k} = \mathbb{D}_{\mathtt{WF-KEM}}(\mathbf{c}, \mathbf{crd}) \,\middle|\, \begin{array}{l} ((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^m \leftarrow \mathbb{G}_{\mathtt{WF-KEM}}(1^\kappa, m) \\ (\mathtt{k}, \mathbf{c}) \leftarrow \mathbb{E}_{\mathtt{WF-KEM}}(\mathcal{P}) \\ Q \leftarrow \mathcal{P} \\ \text{Parse } (X_{i_1}, \dots, X_{i_k}) \leftarrow Q \\ [\mathbf{crd}]_j \leftarrow \mathbb{X}_{\mathtt{WF-KEM}}(X_{i_j}, \mathtt{Msk}_{i_j}), 1 \le j \le k \end{array} \right) = 1.$$

The indistinguishability games against chosen credential and ciphertext attacks for a WF-KEM are defined as follows. As in [1] we call this notion *recipient security.*

```
$(m,n)$-IND-$\mathtt{atk}$
 1. $((\mathtt{Msk}_i,\mathtt{Mpk}_i))_{i=1}^m \leftarrow \mathbb{G}_{\mathtt{WF-KEM}}(1^\kappa,m)$
 2. $(s,\mathcal{P}^*) \leftarrow A_1^{\mathcal{O}_1}(\mathtt{Mpk}_1,\ldots,\mathtt{Mpk}_m)$
 3. $\mathtt{k}_0 \leftarrow \mathbb{K}_{\mathtt{WF-KEM}}$
 4. $(\mathtt{k}_1,\mathbf{c}^*) \leftarrow \mathbb{E}_{\mathtt{WF-KEM}}(\mathcal{P}^*)$
 5. $b \leftarrow \{0,1\}$
 6. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathtt{k}_b,\mathbf{c}^*,s)$
```

$$\mathrm{Adv}_{\mathtt{WF-KEM}}^{(m,n)-\mathtt{IND-atk}}(A) := |\Pr[b'=b]-1/2|.$$

Here $\mathcal{P}^*$ must be on n terms; $\mathcal{O}_1$ and $\mathcal{O}_2$ contain credential extraction and decapsulation oracles subject to the following restrictions:

- The set of queries that the adversary makes to the credential extraction oracle must not form a qualifying set of $\mathcal{P}^*$.
- The adversary cannot query the decapsulation oracle on $\mathbf{c}^*$.

We distinguish adaptive ($\mathtt{atk} = \mathtt{CCCA}$) and non-adaptive ($\mathtt{atk} = \mathtt{CCCA}^-$) attacks. The difference is that in non-adaptive attacks, the adversary is not allowed to query the extraction oracle on any $X = (\mathtt{ID},\mathtt{Mpk})$ with $X \in P^*$ in the second stage of the game.

A WF-KEM scheme is called IND-CCCA (IND-CCCA$^-$) secure if all PPT attackers have negligible advantage in the above game as a function of the security parameter.

Note that WF-KEMs are intrinsically multi-user, as anyone who is able to obtain a qualifying set of credentials will be capable of decapsulating. However, in most practical cases this probably will not be the case, as the credential policy term semantics will include the intended recipient's identity. This is related to another important characteristic of WF-KEMs. Any colluding set of TAs who can produce a qualifying set of credentials are also able to invert the encapsulation, and this means that a WF-KEM is not escrow-free.

4.3 KEMs Supporting Escrow-Free Cryptographic Workflow

The notion of a *KEM supporting escrow-free cryptographic workflow* (EFWF-KEM) implies modifying the previous primitive to remove recipient ambiguity. We follow an approach similar to [1] and [2] whereby the primitive is extended to include a recipient public and private key pair.

EFWF-KEMs are defined through five PT algorithms. Four of these algorithms are analogous to those defined for WF-KEMs. In addition to these we add an extra user key generation algorithm:

- $\mathbb{G}_{\mathtt{EFWF-KEM}}(1^\kappa,m)$: This is the probabilistic authority key generation algorithm which on input of a security parameter 1^κ outputs m authority secret/public key pairs $((\mathtt{Msk}_i,\mathtt{Mpk}_i))_{i=1}^m$, as well as the descriptions of the key, randomness and ciphertext spaces. These are denoted by $\mathbb{K}_{\mathtt{EFWF-KEM}}$, $\mathbb{R}_{\mathtt{EFWF-KEM}}$ and $\mathbb{C}_{\mathtt{EFWF-KEM}}$, respectively.

- $\mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-KEM}}(1^\kappa)$: This is the probabilistic user key generation algorithm which on input of the security parameter 1^κ outputs a private/public key pair $(\mathtt{SK}, \mathtt{PK})$.
- $\mathbb{X}_{\mathtt{EFWF-KEM}}(X, \mathtt{Msk})$: This is the probabilistic credential extraction algorithm which on input of a policy term X, consisting of a policy identifier/authority public key pair $(\mathtt{ID}, \mathtt{Mpk})$, and the secret key $\mathtt{Msk}$ corresponding to $\mathtt{Mpk}$, outputs a pair $(\mathtt{crd}, X)$ which we call a *credential*.
- $\mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}, \mathtt{PK})$: This is the probabilistic key encapsulation algorithm which on input of an access structure $\mathcal{P}$ on n policy terms $P = \{X_1, \ldots, X_n\}$ and a public key $\mathtt{PK}$, outputs a pair $(\mathtt{k}, \mathbf{c})$ where $\mathtt{k} \in \mathbb{K}_{\mathtt{EFWF-KEM}}$ and $\mathbf{c}$ is an encapsulation of $\mathtt{k}$.
- $\mathbb{D}_{\mathtt{EFWF-KEM}}(\mathbf{c}, \mathbf{crd}, \mathtt{SK})$: The deterministic decapsulation algorithm which on input of an encapsulation $\mathbf{c}$, a list of k credentials $\mathbf{crd}$, and a secret key $\mathtt{SK}$ outputs a key or a failure symbol $\perp$.

An EFWF-KEM scheme is called *sound* if for every policy $\mathcal{P}$ on n terms with $m, n \in \mathbb{N}$ we have:

$$\Pr\left(\mathtt{k} = \mathbb{D}_{\mathtt{EFWF-KEM}}(\mathbf{c}, \mathbf{crd}, \mathtt{SK}) \,\middle|\, \begin{array}{l} ((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^{m} \leftarrow \mathbb{G}_{\mathtt{EFWF-KEM}}(1^\kappa, m) \\ (\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-KEM}}(1^\kappa) \\ (\mathtt{k}, \mathbf{c}) \leftarrow \mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}, \mathtt{PK}) \\ Q \leftarrow \mathcal{P}; \text{Parse } (X_{i_1}, \ldots, X_{i_k}) \leftarrow Q \\ [\mathbf{crd}]_j \leftarrow \mathbb{X}_{\mathtt{EFWF-KEM}}(X_{i_j}, \mathtt{Msk}_{i_j}), 1 \leq j \leq k \end{array} \right) = 1.$$

Recipient security for an EFWF-KEM is defined through a game very similar to that presented for a WF-KEM. The only difference is that here the adversary is provided with a user key pair which is generated at the beginning of the game. This captures the notion that even the user who knows the private key must possess a qualifying set of credentials to decapsulate. The game is specified below on the left. Again, $\mathcal{P}^*$ must be on at most n terms; the $\mathcal{O}_1$ and $\mathcal{O}_2$ oracles are exactly as in the previous game for adaptive ($\mathtt{atk} = \mathtt{CCCA}$) and non-adaptive chosen credential attacks ($\mathtt{atk} = \mathtt{CCCA}^-$).

```
$(m, n)$-IND-$\mathtt{atk}$
 1. $((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^{m} \leftarrow \mathbb{G}_{\mathtt{EFWF-KEM}}(1^\kappa)$
 2. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-KEM}}(1^\kappa)$
 3. $(s, \mathcal{P}^*) \leftarrow A_1^{\mathcal{O}_1}((\mathtt{Mpk}_i)_{i=1}^{m}, \mathtt{SK}, \mathtt{PK})$
 4. $\mathtt{k}_0 \leftarrow \mathbb{K}_{\mathtt{EFWF-KEM}}$
 5. $(\mathtt{k}_1, \mathbf{c}^*) \leftarrow \mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}^*, \mathtt{PK})$
 6. $b \leftarrow \{0, 1\}$
 7. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathtt{k}_b, \mathbf{c}^*, s)$
```

```
$(m, n)$-IND-CCA2
 1. $((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^{m} \leftarrow \mathbb{G}_{\mathtt{EFWF-KEM}}(1^\kappa)$
 2. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-KEM}}(1^\kappa)$
 3. $(s, \mathcal{P}^*) \leftarrow A_1^{\mathcal{O}_1}((\mathtt{Msk}_i, \mathtt{Mpk}_i)_{i=1}^{m}, \mathtt{PK})$
 4. $\mathtt{k}_0 \leftarrow \mathbb{K}_{\mathtt{EFWF-KEM}}$
 5. $(\mathtt{k}_1, \mathbf{c}^*) \leftarrow \mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}^*, \mathtt{PK})$
 6. $b \leftarrow \{0, 1\}$
 7. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathtt{k}_b, \mathbf{c}^*, s)$
```

To capture escrow-freeness, we follow the approach in [1] and define *external security* through the indistinguishability game shown above on the right. Note that the adversary controls everything except the user secret key. Here $\mathcal{P}^*$ must be on n terms; $\mathcal{O}_1$ and $\mathcal{O}_2$ denote a decapsulation oracle subject to the restriction that the adversary cannot query it on $\mathbf{c}^*$. An EFWF-KEM scheme is called IND-CCCA (IND-CCCA$^-$) and IND-CCA2 secure if all PPT attackers have negligible advantage in the above games as a function of the security parameter, where advantages are defined as

$$\mathrm{Adv}^{(m,n)-\mathtt{IND-atk}}_{\mathtt{EFWF-KEM}}(A) := |\Pr[b' = b] - 1/2|,$$
$$\mathrm{Adv}^{(m,n)-\mathtt{IND-CCA2}}_{\mathtt{EFWF-KEM}}(A) := |\Pr[b' = b] - 1/2|.$$

4.4 Hybrid Encryption Supporting Cryptographic Workflow

The concept and security model of an encryption scheme supporting escrow-free cryptographic workflow (EFWF-ENC), as proposed in [1], are defined in a very similar manner to an EFWF-KEM. We refer the reader to Appendix A for the details. Using an EFWF-KEM and a standard DEM with compatible key spaces, one can construct a hybrid encryption scheme supporting escrow-free cryptographic workflow in the usual way:

```
$\mathbb{E}_{\mathtt{EFWF-ENC}}(\mathtt{m}, \mathcal{P}, \mathtt{PK})$
  – $(\mathtt{k}, \bar{\mathbf{c}}) \leftarrow \mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}, \mathtt{PK})$
  – $\mathtt{c} \leftarrow \mathbb{E}_{\mathtt{DEM}}(\mathtt{m}, \mathtt{k})$
  – $\mathbf{c} \leftarrow (\bar{\mathbf{c}}, \mathtt{c})$
  – Return $\mathbf{c}$
```

```
$\mathbb{D}_{\mathtt{EFWF-ENC}}(\mathbf{c}, \mathbf{crd}, \mathtt{SK})$
  – $(\bar{\mathbf{c}}, \mathtt{c}) \leftarrow \mathbf{c}$
  – $\mathtt{k} \leftarrow \mathbb{D}_{\mathtt{EFWF-KEM}}(\bar{\mathbf{c}}, \mathbf{crd}, \mathtt{SK})$
  – If $\mathtt{k} = \perp$ then return $\perp$
  – $\mathtt{m} \leftarrow \mathbb{D}_{\mathtt{DEM}}(\mathtt{c}, \mathtt{k})$
  – Return $\mathtt{m}$
```

In Appendix B we also prove the following theorem relating the security of this hybrid encryption scheme to that of its EFWF-KEM and DEM components. We use a technique similar to that in [14]. A similar result holds for non-escrow free primitives.

Theorem 1. *The hybrid EFWF-ENC scheme as constructed above is secure in the recipient and external security models if the underlying EFWF-KEM and DEM are secure. More precisely, for* $\mathtt{atk} \in \{\mathtt{CCCA}, \mathtt{CCCA}^-\}$ *we have:*

$$\mathrm{Adv}^{(m,n)-\mathtt{IND-atk}}_{\mathtt{EFWF-ENC}}(A) \leq 2 \cdot \mathrm{Adv}^{(m,n)-\mathtt{IND-atk}}_{\mathtt{EFWF-KEM}}(B_1) + \mathrm{Adv}^{\mathtt{FG-CCA}}_{\mathtt{DEM}}(B_2),$$

$$\mathrm{Adv}^{(m,n)-\mathtt{IND-CCA2}}_{\mathtt{EFWF-ENC}}(A) \leq 2 \cdot \mathrm{Adv}^{(m,n)-\mathtt{IND-CCA2}}_{\mathtt{EFWF-KEM}}(B_1) + \mathrm{Adv}^{\mathtt{FG-CCA}}_{\mathtt{DEM}}(B_2).$$

5 Generic Constructions

5.1 A WF-KEM Construction

We first present a construction of a WF-KEM using an IBE, a secret sharing scheme and a one-time signature scheme.

The authority key generation and credential extraction algorithms of the resulting WF-KEM are direct adaptations of the master key generation and secret key extraction algorithms of the underlying IBE:

- $\mathbb{G}_{\mathtt{WF-KEM}}(1^\kappa, m)$: Runs the $\mathbb{G}_{\mathtt{IBE}}(1^\kappa)$ algorithm m times obtaining $(\mathtt{Mpk}, \mathtt{Msk})$. The key space is $\mathbb{K}_{\mathtt{WF-KEM}} = \{0,1\}^\kappa$.
- $\mathbb{X}_{\mathtt{WF-KEM}}(X, \mathtt{Msk})$: Parses X to get $(\mathtt{ID}, \mathtt{Mpk})$, extracts $\mathtt{crd} = \mathbb{X}_{\mathtt{IBE}}(\mathtt{ID}, \mathtt{Msk})$ and returns $(\mathtt{crd}, X)$.

The encapsulation and decapsulation algorithms are as follows.

```
$\mathbb{E}_{\mathtt{WF-KEM}}(\mathcal{P})$
  – $(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathbb{G}_{\mathtt{OTS}}(1^\kappa)$
  – $\mathtt{k} \leftarrow \mathbb{K}_{\mathtt{WF-KEM}}$
  – $(\mathbf{shr}, \mathtt{aux}) \leftarrow \mathbb{S}(1^\kappa, \mathtt{k}, \mathcal{P})$
  – For $j = 1, \ldots, n$ do
      $(\mathtt{ID}, \mathtt{Mpk}) \leftarrow X_j$
      $\mathtt{c}_j \leftarrow \mathbb{E}_{\mathtt{IBE}}([\mathbf{shr}]_j || \mathtt{vk}, \mathtt{ID}, \mathtt{Mpk})$
  – $\mathbf{c} \leftarrow (\mathtt{c}_1, \ldots, \mathtt{c}_n, \mathtt{vk}, \mathtt{aux}, \mathcal{P})$
  – $\sigma \leftarrow \mathtt{Sig}(\mathbf{c}, \mathtt{sk})$
  – Return $(\mathtt{k}, \mathbf{c}||\sigma)$
```

```
$\mathbb{D}_{\mathtt{WF-KEM}}(\mathbf{c}||\sigma, \mathbf{crd})$
  – $(\mathtt{c}_1, \ldots, \mathtt{c}_n, \mathtt{vk}, \mathtt{aux}, \mathcal{P}) \leftarrow \mathbf{c}$
  – If $\mathtt{Ver}(\mathbf{c}, \sigma, \mathtt{vk}) \neq 1$ return $\perp$
  – For $j = 1, \ldots, k$ do
      $(\mathtt{crd}, X) \leftarrow [\mathbf{crd}]_j$
      Find $\mathtt{c}_i$ corresponding to $X$
      $([\mathbf{shr}]_j || \mathtt{vk}_j) \leftarrow \mathbb{D}_{\mathtt{IBE}}(\mathtt{c}_i, \mathtt{crd})$
      If $([\mathbf{shr}]_j || \mathtt{vk}_j) = \perp$ return $\perp$
      If $\mathtt{vk}_j \neq \mathtt{vk}$ return $\perp$
  – $\mathtt{k} \leftarrow \mathbb{S}^{-1}(\mathbf{shr}, \mathtt{aux})$
  – If $\mathtt{k} = \perp$ return $\perp$
  – Return $\mathtt{k}$
```

Note that, similarly to what is done in [16] for multiple encryption in the public-key setting, one could use an IBE primitive modified to include non-malleable public labels to bind $\mathtt{vk}$ to each individual $\mathtt{c}_j$. We chose not to do this so that we could base our construction on the more standard IBE primitive and security model. The security of the above construction is captured via the following theorem which is proved in Appendix C.

Theorem 2. *The above construction is (m, n)-IND-CCCA secure if the underlying IBE is IND-CCA2 secure, the OTS is UF secure, and the secret sharing scheme is information theoretically secure. More precisely we have:*

$$\mathrm{Adv}^{\mathtt{IND-CCCA}}_{\mathtt{WF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2mn^2 \cdot \mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{IBE}}(B_2).$$

The best result we obtain in the standard model for computational secret sharing schemes is the following. In Section 6 we explain why this is the case.

Theorem 3. *The above construction is (m, n)-IND-CCCA$^-$ secure if the underlying IBE is IND-CCA2 secure, the OTS is UF secure, and the secret sharing scheme is IND-SSA secure. More precisely we have:*

$$\mathrm{Adv}^{\mathtt{IND-CCCA}^-}_{\mathtt{WF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2mn^2 \cdot \mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{IBE}}(B_2) + \mathrm{Adv}^{\mathtt{IND-SSA}}_{\mathtt{SS}}(B_3).$$

Proof. (*Sketch*) The proof is very similar to the one included in Appendix C for Theorem 2. However, in this case, we know exactly which credentials the adversary has extracted during the first stage, and it is unable to extract credentials related to the challenge in stage two. This makes it possible to show that, if the IBE scheme is IND-CCA2 secure, the adversary's advantage changes negligibly if we change the ciphertext components to which the adversary has no access by encrypting random bit strings of appropriate length. Once in this game environment, the adversary's advantage can then be used to directly win the IND-SSA game against the secret sharing scheme. The simulator selects the shares that the adversary will be recovering from the external IND-SSA game when it is about to construct the challenge. Since all the other ciphertext components contain random data, any advantage the adversary obtains must come from attacking the secret sharing scheme. □

5.2 An EFWF-KEM Construction

We now extend the previous generic construction to achieve escrow-freeness. We build an EFWF-KEM using an additional component: a PKE scheme. The authority key generation and credential extraction algorithms are as in the WF-KEM construction. The user key generation algorithm is that of the underlying PKE. Finally, the encapsulation and decapsulation algorithms are:

```
$\mathbb{E}_{\mathtt{EFWF-KEM}}(\mathcal{P}, \mathtt{PK})$
  – $(\mathtt{vk}, \mathtt{sk}) \leftarrow \mathbb{G}_{\mathtt{OTS}}(1^{\kappa})$
  – $\mathtt{k}_1, \mathtt{k}_2 \leftarrow \mathbb{K}_{\mathtt{EFWF-KEM}}$
  – $(\mathbf{shr}, \mathtt{aux}) \leftarrow \mathbb{S}(1^{\kappa}, \mathtt{k}_1, \mathcal{P})$
  – $\bar{\mathtt{c}} \leftarrow \mathbb{E}_{\mathtt{PKE}}(\mathtt{k}_2 || \mathtt{vk}, \mathtt{PK})$
  – For $j = 1, \ldots, n$ do
      $(\mathtt{ID}, \mathtt{Mpk}) \leftarrow X_j$
      $\mathtt{c}_j \leftarrow \mathbb{E}_{\mathtt{IBE}}([\mathbf{shr}]_j || \mathtt{vk}, \mathtt{ID}, \mathtt{Mpk})$
  – $\mathbf{c} \leftarrow (\bar{\mathtt{c}}, \mathtt{c}_1, \ldots, \mathtt{c}_n, \mathtt{vk}, \mathtt{aux}, \mathcal{P})$
  – $\sigma \leftarrow \mathtt{Sig}(\mathbf{c}, \mathtt{sk})$
  – Return $(\mathtt{k}_1 \oplus \mathtt{k}_2, \mathbf{c} || \sigma)$
```

```
$\mathbb{D}_{\mathtt{EFWF-KEM}}(\mathbf{c} || \sigma, \mathbf{crd}, \mathtt{SK})$
  – $(\bar{\mathtt{c}}, \mathtt{c}_1, \ldots, \mathtt{c}_n, \mathtt{vk}, \mathtt{aux}, \mathcal{P}) \leftarrow \mathbf{c}$
  – If $\mathtt{Ver}(\mathbf{c}, \sigma, \mathtt{vk}) \neq 1$ return $\perp$
  – For $j = 1, \ldots, k$ do
      $(\mathtt{crd}, X) \leftarrow [\mathbf{crd}]_j$
      Find $\mathtt{c}_i$ corresponding to $X$
      $([\mathbf{shr}]_j || \mathtt{vk}_j) \leftarrow \mathbb{D}_{\mathtt{IBE}}(\mathtt{c}_i, \mathtt{crd})$
      If $([\mathbf{shr}]_j || \mathtt{vk}_j) = \perp$ return $\perp$
      If $\mathtt{vk}_j \neq \mathtt{vk}$ return $\perp$
  – $\mathtt{k}_1 \leftarrow \mathbb{S}^{-1}(\mathbf{shr}, \mathtt{aux})$
  – $\mathtt{k}_2 \leftarrow \mathbb{D}_{\mathtt{PKE}}(\bar{\mathtt{c}}, \mathtt{SK})$
  – If $\mathtt{k}_1 = \perp$ or $\mathtt{k}_2 = \perp$ return $\perp$
  – Return $\mathtt{k}_1 \oplus \mathtt{k}_2$
```

Again we have two security results which depend on the security provided by the underlying secret sharing scheme. The following theorem is proved in Appendix D.

Theorem 4. *The above EFWF-KEM construction is (m,n)-IND-CCCA and (m,n)-IND-CCA2 secure if the underlying PKE and IBE are IND-CCA2 secure, the OTS is UF secure, and the secret sharing scheme is information-theoretically secure. More precisely we have:*

$$\mathrm{Adv}^{\mathtt{IND-CCCA}}_{\mathtt{EFWF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2mn^2 \cdot \mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{IBE}}(B_2),$$

$$\mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{EFWF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2\mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{PKE}}(B_2).$$

Theorem 5. *The above EFWF-KEM construction is (m,n)-IND-CCCA^{-} and (m,n)-IND-CCA2 secure if the underlying PKE is IND-CCA2 secure, the underlying IBE is IND-CCA2 secure, the OTS is UF secure, and the secret sharing scheme is IND-SSA secure. More precisely we have:*

$$\mathrm{Adv}^{\mathtt{IND-CCCA}^{-}}_{\mathtt{EFWF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2mn^2 \cdot \mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{IBE}}(B_2) + \mathrm{Adv}^{\mathtt{IND-SSA}}_{\mathtt{SS}}(B_3),$$

$$\mathrm{Adv}_{\mathtt{EFWF-KEM}}^{\mathtt{IND-CCA2}}(A) \leq \mathrm{Adv}_{\mathtt{OTS}}^{\mathtt{UF}}(B_1) + 2\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-CCA2}}(B_2).$$

Proof. (*Sketch*) The only difference introduced by allowing for a computational secret sharing scheme resides on the credential security result, as recipient security is guaranteed by the OTS and PKE schemes. The same argument presented for the WF-KEM construction in Theorem 3 applies here. □

6 Discussion

The main contribution in this work is the fact that, through the generic constructions that we propose, and using underlying components which achieve the required levels of security in the standard model, we obtain the first WF-KEM and EFWF-KEM schemes provably secure in the standard model.

There are, however, other interesting aspects to the results presented in the previous sections, which we now discuss.

Relation with the original construction in [1] : The concrete EFWF-KEM scheme in [1] is originally defined as a full encryption scheme, although internally it is structured as a KEM-DEM construction. The basic building block in the KEM part is a weak version of the IBE scheme by Boneh and Franklin [7]. Chosen ciphertext security is achieved globally through a transformation akin to that used in the KEM constructions in [15], which is valid in the random oracle model. We require fully chosen ciphertext secure individual components, and the way we achieve global CCA2 security in the standard model comes from the IBE to PKE transformation in [10], adapted to multiple encryption in [16].

Our constructions do inherit the combination of a secret sharing scheme, an IBE scheme and a PKE scheme. However, if we allow for computational secret sharing, then we can only achieve IND-CCCA$^-$ security. This is true even if the underlying secret sharing scheme tolerates adaptive chosen share attacks. This is the main difference between the security of our construction and that in [1]. Intuitively this can be explained as follows. Using the RO heuristic one can perform a *late binding* between challenge share values and the challenge ciphertext. This makes it possible to construct the challenge without explicitly knowing the shares, and directly map the adversary's credential extraction queries to external calls to a share extraction oracle.

The standard model does not allow the same proof strategy, so we cannot prove the security of our constructions against adaptive credential extracting attackers unless we adopt perfect secret sharing. This will only be an issue in terms of the overall efficiency of the constructions, which we discuss below.

Finally, it is interesting to note the very effective application of the randomness reuse paradigm [4] in [1] to achieve impressive computational and ciphertext length savings.

Relation with multiple encryption : This work builds on the general results by Dodis and Katz [16] for chosen ciphertext security of multiple encryption.

However our constructions require that we extend these results in three different aspects: (1) to consider adaptive user corruption attacks, (2) to consider generalised access structures and (3) we require a mix of identity-based and public-key encryption techniques. Our results imply that equivalent extensions can be derived in the context of generic multiple encryption.

Relation with certificateless encryption : We will not explore this connection in detail due to space constraints. However, we do note the similarity between the security models of a CL-KEM scheme [6] and the EFWF-KEM security models introduced. This similarity implies that a simplified version of our construction considering only one credential and a single authority can be seen as a CL-KEM scheme which can be proved IND-CCA2 secure against Type I- and Type II adversaries [6].

Efficiency considerations : We analyse the efficiency of our constructions by looking at the computational load and ciphertext length that they produce. A high level analysis shows that the computational weight associated with encapsulation and decapsulation is that of sharing the secret key, encrypting the n shares using the IBE scheme, possibly encrypting another secret key with the PKE scheme, and generating a one-time signature. The corresponding ciphertexts include the public sharing information, n IBE ciphertexts, possibly one PKE ciphertext, the OTS verification key and a signature string.

An obvious way to optimise the end-result is to choose underlying components which are themselves efficient. For example, adopting the IBE scheme of Sakai and Kasahara [11] one obtains a solution which is computationally more efficient than the original construction in [1]. However, there are three techniques which can further improve the efficiency of our constructions.

The enhanced IBE to PKE transformation proposed in [8], which replaces the OTS component by a MAC and a weak form of commitment has also been adapted to achieve chosen ciphertext security in [16] for a weak form of multiple encryption. It turns out that this weak form of multiple encryption is sufficient to allow an extension to WF-KEMs similar to what we achieved with the OTS-based technique. We chose not to include these results in this paper as they lead to more involved proofs and they are less intuitive.

The randomness reuse paradigm [4] can also be applied in this context, although to the best of our knowledge there is currently no IBE scheme which is IND-CCA2 secure in the standard model, and which allows reuse of randomness. This, in itself, is an interesting open problem. However, if we settle for the fully secure version of the Boneh and Franklin IBE scheme, then we can obtain bandwidth and computational (point multiplication) savings by re-using the first component in all IBE ciphertexts. Further improvements may be attainable by re-using the same randomness in the PKE component as in [1].

Our constructions can be easily adapted to work with IBE and PKE schemes extended to take labels as additional parameters, and bind them non-malleably to the ciphertext. This adaptation reduces to using the OTS verification key as the label parameter. Potential benefits of this would arise from labelled IBE or

PKE schemes which achieve this functionality more efficiently than the direct non-malleable labelling that we adopted in our constructions.

As a final note on efficiency, we look at the potential benefits of using a computational secret sharing scheme rather than a perfect secret sharing scheme. The main advantage in this is to obtain share sizes which are smaller than the shared secret, which is important when the secret is large. For example, the scheme in [21] uses a perfect secret sharing scheme as an underlying component to split an auxiliary secret key. This key is then used to encrypt the results of partitioning the (large) secret using an information dispersal algorithm. This provides share sizes which asymptotically approach the optimal $|S|/n$ by detaching the size of the (large) secret from the input to the perfect secret sharing scheme. In our case this is an invalid argument, as the secrets we share are themselves secret keys.

7 Acknowledgements

The authors would like to thank Nigel Smart for his helpful comments. Part of this work was carried out when the second author visited University of Minho in Portugal, funded by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

References

1. S.S. Al-Riyami, J. Malone-Lee and N.P. Smart. Escrow-Free Encryption Supporting Cryptographic Workflow. *Cryptology ePrint Archive*, Report 2004/258. 2004.
2. S.S. Al-Riyami and K.G. Paterson. Certificateless Public-Key Cryptography. *Advances in Cryptology - ASIACRYPT 2003*, LNCS 2894:452–473. Springer-Verlag, 2003.
3. M. Bellare, A. Boldyreva and S. Micali. Public-Key Encryption in a Multi-User Setting: Security Proofs and Improvements. *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807:259–274. Springer-Verlag, 2000.
4. M. Bellare, A. Boldyreva and J. Staddon. Randomness Re-Use in Multi-Recipient Encryption Schemes. *Public Key Cryptography - PKC 2003*, LNCS 2567:85–99. Springer-Verlag, 2003.
5. J. Benaloh and J. Leichter. Generalized Secret Sharing and Monotone Functions. *Advances in Cryptology - CRYPTO '88*, LNCS 403:27–35. Springer-Verlag, 1990.
6. K. Bentahar, P. Farshim, J. Malone-Lee and N.P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. *Cryptology ePrint Archive*, Report 2005/058, 2005.
7. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32:586–615. 2003.
8. D. Boneh and J. Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. *Cryptology ePrint Archive*, Report 2004/261, 2004.
9. R.W. Bradshaw, J.E. Holt and K.E. Seamons. Concealing Complex Policies with Hidden Credentials. *11th ACM Conference on Computer and Communications Security*, 2004.

10. R. Canetti, S. Halevi and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *Cryptology ePrint Archive*, Report 2003/182, 2003.
11. L. Chen and Z. Cheng. Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. *Cryptography and Coding*, LNCS 3796:442–459. Springer-Verlag, 2005.
12. L. Chen and K. Harrison. Multiple Trusted Authorities in Identifier Based Cryptography from Pairings on Elliptic Curves. *Technical Report, HPL-2003-48*, HP Laboratories, 2003.
13. L. Chen, K. Harrison, D. Soldera and N.P. Smart. Applications of Multiple Trusted Authorities in Pairing Based Cryptosystems. *Proceedings InfraSec 2002*, LNCS 2437:260–275. Springer-Verlag, 2002.
14. R. Cramer and V. Shoup. A Practical Public-Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. *Advances in Cryptology - CRYPTO '98*, LNCS 1462:13–25. Springer-Verlag, 1998.
15. A.W. Dent. A Designer's Guide to KEMs. *Coding and Cryptography*, LNCS 2898:133–151. Springer-Verlag, 2003.
16. Y. Dodis and J. Katz. Chosen-Ciphertext Security of Multiple Encryption. *TCC 2005*, LNCS 3378:188–209. Springer-Verlag, 2005.
17. C. Gentry. Practical identity-based encryption without random oracles. *Advances in Cryptology - EUROCRYPT 2006*, LNCS 4004:445–464. Springer-Verlag, 2006.
18. J. Herranz and D. Hofheinz and E. Kiltz. KEM/DEM: Necessary and Sufficient Conditions for Secure Hybrid Encryption. *Cryptology ePrint Archive*, Report 2006/265. 2006.
19. J.E. Holt, R.W. Bradshaw, K.E. Seamons and H. Orman. Hidden Credentials. *2nd ACM Workshop on Privacy in the Electronic Society*, pp. 1–8, 2003.
20. E. Kiltz. Chosen-Ciphertext Secure Identity-Based Encryption in the Standard Model with short Ciphertexts. *Cryptology ePrint Archive*, Report 2006/122, 2006.
21. H. Krawczyk. Secret Sharing Made Short. *Proceedings of Crypto'93 - Advances in Cryptology*, LNCS. Springer-Verlag, 1993.
22. W. Nagao, Y. Manabe and T. Okamoto. On the Equivalence of Several Security Notions of Key Encapsulation Mechanism. *Cryptology ePrint Archive*, Report 2006/268. 2006.
23. K.G. Paterson. Cryptography from Pairings: A Snapshot of Current Research. *Information Security Technical Report*, 7:41–54, 2002.
24. R. Sakai and M. Kasahara. ID-Based Cryptosystems with Pairing on Elliptic Curve. *Cryptology ePrint Archive*, Report 2003/054, 2003.
25. A. Shamir. How to Share a Secret. *Communications of the ACM*, 22:612–613, 1979.
26. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. *Proceedings of CRYPTO '84 on Advances in Cryptology*, LNCS 196:47–53. Springer-Verlag, 1985.
27. N.P. Smart. Access Control Using Pairing Based Cryptography. *Topics in Cryptology - CT-RSA 2003*, LNCS 2612:111–121. Springer-Verlag, 2003.
28. B.R. Waters. Efficient Identity-Based Encryption Without Random Oracles. *Cryptology ePrint Archive*, Report 2004/180, 2004.

Appendix A – Encryption Schemes Supporting Escrow-Free Cryptographic Workflow

Following [1], an encryption scheme supporting escrow-free cryptographic workflow (EFWF-ENC) is defined via five PT algorithms:

- $\mathbb{G}_{\mathtt{EFWF-ENC}}(1^\kappa, m)$: This is the probabilistic authority key generation algorithm which on input of a security parameter 1^κ and an integer m outputs an m-tuple $((\mathtt{Msk}_1, \mathtt{Mpk}_1), \ldots, (\mathtt{Msk}_m, \mathtt{Mpk}_m))$ of authority secret and public keys.
- $\mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-ENC}}(1^\kappa)$: This is the probabilistic user key generation algorithm which on input of a security parameter 1^κ outputs a secret/public key pair $(\mathtt{SK}, \mathtt{PK})$.
- $\mathbb{X}_{\mathtt{EFWF-ENC}}(X, \mathtt{Msk})$: This is the probabilistic credential extraction algorithm which on input of a policy term X, consisting of a policy identifier/authority public key pair $(\mathtt{ID}, \mathtt{Mpk})$, and the secret key $\mathtt{Msk}$ corresponding to $\mathtt{Mpk}$, outputs the pair $(\mathtt{crd}, X)$ which we call a *credential.*
- $\mathbb{E}_{\mathtt{EFWF-ENC}}(\mathtt{m}, \mathcal{P}, \mathtt{PK})$: This is the probabilistic encryption algorithm which on input of a message $\mathtt{m}$, an access structure $\mathcal{P}$ and a public key $\mathtt{PK}$, outputs a ciphertext $\mathbf{c}$.
- $\mathbb{D}_{\mathtt{EFWF-ENC}}(\mathbf{c}, \mathbf{crd}, \mathtt{SK})$: The deterministic decryption algorithm which on input of a ciphertext $\mathbf{c}$, a list of credentials $\mathbf{crd}$ and a secret key $\mathtt{SK}$, outputs a message $\mathtt{m}$ or a failure symbol $\perp$.

An EFWF-ENC scheme is *sound* if for every policy $\mathcal{P}$ on n terms with $m, n \in \mathbb{N}$ and every message $\mathtt{m}$ we have:

$$\Pr\left(\mathtt{m} = \mathbb{D}_{\mathtt{EFWF-ENC}}(\mathbf{c}, \mathbf{crd}, \mathtt{SK}) \middle| \begin{array}{l} ((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^m \leftarrow \mathbb{G}_{\mathtt{EFWF-ENC}}(1^\kappa, m) \\ (\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-ENC}}(1^\kappa) \\ \mathbf{c} \leftarrow \mathbb{E}_{\mathtt{EFWF-ENC}}(\mathtt{m}, \mathcal{P}, \mathtt{PK}) \\ Q \leftarrow \mathcal{P}; \text{Parse } (X_{i_1}, \ldots, X_{i_k}) \leftarrow Q \\ [\mathbf{crd}]_j \leftarrow \mathbb{X}_{\mathtt{EFWF-ENC}}(X_{i_j}, \mathtt{Msk}_{i_j}), 1 \le j \le k \end{array}\right) = 1.$$

The recipient security model is defined below. Here, $\mathcal{P}^*$ must be on n terms, and $\mathcal{O}_1$ and $\mathcal{O}_2$ denote credential extraction and decryption oracles subject to the following restrictions. In the CCCA model the set of queries that the adversary makes to the credential extraction oracle should not form a qualifying subset of P^*. In the CCCA$^-$ model there is the additional restriction that the adversary cannot call the extraction oracle on any $X = (\mathtt{ID}, \mathtt{Mpk})$ with $X \in P^*$ in the second stage of the game. Also, the adversary cannot query the decryption oracle on $\mathbf{c}^*$.

```
$(m, n)$-IND-$\mathtt{atk}$
 1. $((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^m \leftarrow \mathbb{G}_{\mathtt{EFWF-ENC}}(1^\kappa, m)$
 2. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-ENC}}(1^\kappa)$
 3. $(s, \mathtt{m}_0, \mathtt{m}_1, \mathcal{P}^*) \leftarrow A_1^{\mathcal{O}_1}((\mathtt{Mpk}_i)_{i=1}^m, \mathtt{SK}, \mathtt{PK})$
 4. $b \leftarrow \{0, 1\}$
 5. $\mathbf{c}^* \leftarrow \mathbb{E}_{\mathtt{EFWF-ENC}}(\mathtt{m}_b, \mathcal{P}^*, \mathtt{PK})$
 6. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathbf{c}^*, s)$
```

$$\mathrm{Adv}_{\mathtt{EFWF-ENC}}^{(m,n)-\mathtt{IND-atk}}(A) := |\Pr[b' = b] - 1/2|.$$

Escrow-freeness, or external security, is captured via the game below. Here, $\mathcal{P}^*$ must be on n terms, and $\mathcal{O}_1$ and $\mathcal{O}_2$ denote a decryption oracle subject to the restriction that the adversary cannot query it on $\mathbf{c}^*$.

```
(m, n)-IND-CCA2
 1. $((\mathtt{Msk}_i, \mathtt{Mpk}_i))_{i=1}^m \leftarrow \mathbb{G}_{\mathtt{EFWF-ENC}}(1^\kappa, m)$
 2. $(\mathtt{SK}, \mathtt{PK}) \leftarrow \mathbb{G}^{\mathtt{U}}_{\mathtt{EFWF-ENC}}(1^\kappa)$
 3. $(s, \mathtt{m}_0, \mathtt{m}_1, \mathcal{P}^*) \leftarrow A_1^{\mathcal{O}_1}((\mathtt{Msk}_i, \mathtt{Mpk}_i)_{i=1}^m, \mathtt{PK})$
 4. $b \leftarrow \{0, 1\}$
 5. $\mathbf{c}^* \leftarrow \mathbb{E}_{\mathtt{EFWF-ENC}}(\mathtt{m}_b, \mathcal{P}^*, \mathtt{PK})$
 6. $b' \leftarrow A_2^{\mathcal{O}_2}(\mathbf{c}^*, s)$
```

$$\mathrm{Adv}_{\mathtt{EFWF-ENC}}^{(m,n)-\mathtt{IND-CCA2}}(A) := |\Pr[b' = b] - 1/2|.$$

Appendix B – Proof of Theorem 1

Proof. The proof is very similar to that in [6]. We give the details for completeness. It is done concurrently for the attack model $\mathtt{atk}$, which is either IND-CCCA2, IND-CCCA1 or IND-CCA2, through a sequence $\mathtt{Game}_0, \mathtt{Game}_1$ and $\mathtt{Game}_2$ of modified attack games.

We fix some notation that we will use throughout. Let $\mathbf{c}^* = (\bar{\mathbf{c}}^*, \mathtt{c}^*)$ be the challenge ciphertext presented to A by its challenge encryption oracle – the oracle that encrypts either $\mathtt{m}_0$ or $\mathtt{m}_1$ according to a bit b. Let $\mathtt{k}^*$ denote the symmetric key used by the challenge encryption oracle in the generation of the challenge ciphertext, or alternatively, the decapsulation of $\bar{\mathbf{c}}^*$ using the credentials associated to $\mathcal{P}^*$ – the policy chosen by the adversary on which it wishes to be challenged – and the public key $\mathtt{PK}$. For any $i = 0, 1, 2$, we let S_i be the event that $b' = b$ in game $\mathtt{Game}_i$, where b is the bit chosen by A's challenge encryption oracle. This probability is taken over the random choices of A and those of A's oracles.

Let $\mathtt{Game}_0$ be the genuine attack game played by A. By definition we have

$$|\Pr[S_0] - 1/2| = \mathrm{Adv}_{\mathtt{EFWF-ENC}}^{\mathtt{atk}}(A).$$

$\mathtt{Game}_0$ is now modified so that whenever $(\bar{\mathbf{c}}^*, \mathtt{c})$ is presented to the decryption oracle after the invocation of the challenge encryption oracle, then the decryption oracle does not use the genuine decryption procedure for the hybrid scheme, instead it uses the key $\mathtt{k}^*$ to decapsulate $\mathtt{c}$ and returns the result to the adversary.

This modification to $\mathtt{Game}_0$ gives us the game $\mathtt{Game}_1$. Games $\mathtt{Game}_0$ and $\mathtt{Game}_1$ are identical – under the soundness condition – and so $\Pr[S_1] = \Pr[S_0]$[5].

We now modify $\mathtt{Game}_1$ by replacing $\mathtt{k}^*$ with a random key $\mathtt{k}'$. With this modification we have the game $\mathtt{Game}_2$. The result then follows from the following two lemmas. □

Lemma 1. *There is a PPT algorithm B_1, whose running time is essentially the same as that of A, such that*

$$|\Pr[S_2] - \Pr[S_1]| = 2\mathrm{Adv}_{\mathtt{EFWF-KEM}}^{\mathtt{atk}}(B_1).$$

[5] We may weaken the soundness definition to allow a negligible failure in decryption, which results in a negligible difference between $\mathtt{Game}_0$ and $\mathtt{Game}_1$.

Proof. To prove this we demonstrate how to construct an adversary B_1 of the KEM to violate the assumed credential or external security. Adversary B_1 is constructed by running adversary A and responding to its queries as follows.

- When A calls any oracle, bar its decryption or challenge encryption oracles, then B_1 simply relays these queries to its own equivalent oracle in the corresponding security game.
- To respond to A's decryption oracle query on $(\bar{\mathbf{c}}, \mathtt{c})$ before A has queried its challenge encryption oracle, B_1 proceeds as follows. It first obtains $\mathtt{k}$ by calling its own decapsulation oracle with $\bar{\mathbf{c}}$. If $\mathtt{k} = \perp$ then B_1 replies to A with $\perp$. Otherwise it proceeds to use $\mathtt{k}$ to decrypt $\mathtt{c}$ and relays the result to A.
- When A calls its challenge encryption oracle with policy $\mathcal{P}^*$ and messages $(\mathtt{m}_0, \mathtt{m}_1)$, B_1 first calls its own challenge encryption oracle with $\mathcal{P}^*$ to obtain $(\mathtt{k}^\dagger, \bar{\mathbf{c}}^*)$, where $\mathtt{k}^\dagger$ is either a random key or the proper key encapsulated in $\bar{\mathbf{c}}^*$. It then chooses a bit d at random and computes $\mathtt{c}^* \leftarrow \mathbb{E}_{\mathtt{DEM}}(\mathtt{m}_d, \mathtt{k}^\dagger)$. Finally, it responds to A with $\mathbf{c}^* \leftarrow (\bar{\mathbf{c}}^*, \mathtt{c}^*)$.
- To respond to A's decryption oracle query on $(\bar{\mathbf{c}}, \mathtt{c})$ after A has queried its challenge encryption oracle, B_1 proceeds as follows. It first obtains $\mathtt{k}$ by calling its own decapsulation oracle with $\bar{\mathbf{c}}$. If $\mathtt{k} = \perp$ then B_1 replies to A with $\perp$. Otherwise it proceeds to use $\mathtt{k}$ to decrypt $\mathtt{c}$ and relays the result to A.
- In the particular case where A's query is of the form $(\bar{\mathbf{c}}^*, \mathtt{c})$, B_1 uses $\mathtt{k}^\dagger$ to decrypt $\mathtt{c}$ and relays the result to A. Note that $\mathtt{k}^\dagger$ is the key used in encryption.

At the end of the simulation, A outputs a bit d'. If $d' = d$, B_1 outputs 1, otherwise it outputs 0.

Let b be the internal bit of B_1's challenge oracle which B_1 seeks to determine and let b' be the bit output by B_1. By construction we see that when $b = 1$, so $\mathtt{k}'$ is the key encapsulated within $\mathbf{c}^*$, A is run exactly as it would be run in $\mathtt{Game}_1$. This means that

$$\Pr[S_1] = \Pr[d' = d | b = 1] = \Pr[b' = 1 | b = 1]$$

where d is A's challenge bit and d' is A's guess. Also, when $b = 0$, so a random $\mathtt{k}'$ is used in the generation of the challenge ciphertext, A is run exactly as it would be in $\mathtt{Game}_2$. This means that

$$\Pr[S_2] = \Pr[d' = d | b = 0] = \Pr[b' = 1 | b = 0].$$

The result follows from the above two equations and the definitions of security for EFWF-KEMs when one observes that

$$\mathrm{Adv}^{\mathtt{atk}}_{\mathtt{EFWF-KEM}}(B_1) = |\Pr[b' = b] - \frac{1}{2}| = \frac{1}{2}|\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]|.$$

□

Lemma 2. *There is a PPT algorithm B_2, whose running time is essentially the same as that of A, such that*

$$|\Pr[S_2] - 1/2| = \mathrm{Adv}_{\mathtt{DEM}}^{\mathtt{FG-CCA}}(B_2).$$

Proof. To construct such a B_2 we simply run A as it would be run in game $\mathtt{Game}_2$. We run the EFWF-KEM's key generation algorithms so we can respond to A's queries before it calls its challenge encryption oracle. When A calls its challenge encryption oracle with identity $\mathcal{P}^*$ and messages $(\mathtt{m}_0, \mathtt{m}_1)$ we simply relay $(\mathtt{m}_0, \mathtt{m}_1)$ to the challenge encryption oracle of B_2 to obtain $\mathtt{c}^*$. We then run the key encapsulation mechanism to obtain $(\mathtt{k}, \bar{\mathbf{c}}^*)$. We now set $\mathbf{c}^* \leftarrow (\bar{\mathbf{c}}^*, \mathtt{c}^*)$ and return it to A. Note that $\mathtt{k}$ is irrelevant here, as the actual key used to create $\mathbf{c}^*$ is a random (unknown) key.

We continue to respond to A's queries as before except if it a makes decryption query $(\bar{\mathbf{c}}^*, \mathtt{c})$. In this instance we query B_2's decryption oracle with $\mathtt{c}$ and relay the response to A. This is needed since we need to follow the rules of $\mathtt{Game}_2$.

At the end of simulation B_2 outputs whatever A outputs. In this simulation A is run by B_2 in exactly the same manner as the former would be run in game $\mathtt{Game}_2$. Moreover, $\Pr[S_2]$ corresponds exactly to the probability that B_2 correctly determines the hidden bit of its challenge encryption oracle since. The result follows. □

Appendix C – Proof of Theorem 2

Proof. We prove the theorem using a sequence of five games $\mathtt{Game}_0, \ldots, \mathtt{Game}_4$. Let A be an adversary against the generic workflow construction. We denote by S_i the event that A guesses the challenge bit correctly in $\mathtt{Game}_i$.

Let $\mathtt{Game}_0$ be the original IND-CCCA attack game. Hence

$$\mathrm{Adv}_{\mathtt{WF-KEM}}^{\mathtt{IND-CCCA}}(A) = |\Pr[S_0] - 1/2|.$$

To obtain $\mathtt{Game}_1$ we introduce a single change: all decapsulation queries where the OTS verification key included in the challenge is reused by the adversary are answered immediately with $\perp$.

We claim that A's probability of success changes negligibly. Let E denote the event that the adversary submits for decapsulation a valid ciphertext $(\mathbf{c}||\sigma)$, different from the challenge ciphertext $(\mathbf{c}^*||\sigma^*)^6$. Given that $\mathtt{Game}_0$ and $\mathtt{Game}_1$ are identical, unless E occurs, we have

$$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[E]$$

To show that this difference is negligible, it suffices to demonstrate that $\Pr[E]$ must be negligible. This follows easily from the observation that any adversary

[6] Note that *different* in this case means a single bit and, in particular, allows the attacker to reuse $\mathbf{c}^*$ or σ^*.

that causes E to occur with non-negligible probability can be used to directly construct an algorithm B_1 which wins the UF game against the OTS scheme with advantage $\Pr[E]$. Therefore

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1).$$

Now we change $\mathtt{Game}_1$ so that decapsulation queries immediately return $\perp$ for all ciphertexts which reuse one or more challenge components c_i^* and corresponding $\mathcal{P}^*$ terms. We call this new game $\mathtt{Game}_2$ and we claim that

$$\Pr[S_1] = \Pr[S_2].$$

To prove this claim, we argue that $\perp$ is the correct decapsulation result for this type of ciphertext in $\mathtt{Game}_1$. To see this, note that all queries which reuse $\mathtt{vk}^*$ are already returning $\perp$, so we only need to consider the case where $\mathtt{vk} \neq \mathtt{vk}^*$. However, by construction, all c_i^* components will return $\mathtt{vk}^*$ when decrypted, causing the decapsulation consistency check to fail.

We obtain $\mathtt{Game}_3$ by changing the way in which the challenge encapsulation is constructed: the c_j challenge components are constructed using shares resulting from a completely random key $\mathtt{k}_2$.

Once again, we claim that A's probability of success changes only negligibly when one moves to $\mathtt{Game}_3$. We prove this claim using a hybrid argument similar to that in [3].

We show that if $|\Pr[S_2] - \Pr[S_3]|$ is non-negligible, then it is possible to build an algorithm B_2 which runs A as a subroutine, interpolates between the two games, and has non-negligible advantage in the IND-CCA2 game that defines the security of the IBE scheme. B_2 works as follows:

- B_2 chooses a random value ℓ in the range $1, \ldots, n$, where n is the number of policy terms.
- B_2 generates the key pairs for $m-1$ credential authorities and obtains the m-th master public key from the external IBE attack game. B_2 randomly permutes these public keys and passes them on to A.
- On input of a challenge policy $\mathcal{P}^*$, B_2 constructs the challenge encapsulation as follows:
  - If the master public key from the external IBE game is not associated with the ℓ-th policy term (event F_1), B_2 terminates. Let X_ℓ be the first component of the ℓ-th policy term, and $\mathtt{ID}_\ell$ the identifier inside it.
  - B_2 chooses three keys $\mathtt{k}_0$, $\mathtt{k}_1$ and $\mathtt{k}_2$ at random, passes $\mathtt{k}_1$ and $\mathtt{k}_2$ together with $\mathcal{P}^*$ to the secret sharing algorithm to obtain $(\mathbf{shr}_1^*, \mathtt{aux}_1^*)$ and $(\mathbf{shr}_2^*, \mathtt{aux}_2^*)$.
  - B_2 runs $\mathbb{G}_{\mathtt{OTS}}$ to obtain $(\mathtt{vk}^*, \mathtt{sk}^*)$.
  - For shares $1, \ldots, \ell-1$, B_2 constructs c_j^* using $[\mathbf{shr}_1^*]_j || \mathtt{vk}^*$.
  - For the ℓ-th share, B_2 calls the external challenge oracle on $\mathtt{ID}_\ell$ with $(\mathtt{m}_0, \mathtt{m}_1)$, where $\mathtt{m}_0 = [\mathbf{shr}_1^*]_\ell || \mathtt{vk}^*$ and $\mathtt{m}_1 = [\mathbf{shr}_2^*]_\ell || \mathtt{vk}^*$.

  - For all remaining shares, B_2 constructs c_j^* using $[\mathbf{shr}_2^*]_j||\mathtt{vk}^*$.
  - B_2 now generates a random bit b and provides $\mathtt{k}_b$ to the adversary, along with the challenge ciphertext.
- Credential extraction queries are handled as follows:
  - The knowledge of the master secret keys on $m-1$ of the authorities allows B_2 to directly answer most extraction queries using the IBE extraction algorithm.
  - For credentials associated with the authority from the external security game, B_2 will call the secret key extraction oracle provided in that game.
  - Algorithm B_2 will terminate if A chooses to extract the secret key associated with X_ℓ (event F_2), as this would be an invalid query in the IBE game.
- Decapsulation queries are answered as follows:
  - The necessary credentials are obtained by running the credential extraction simulation algorithm above.
  - The exception is X_ℓ in the challenge, for which B_2 simply calls the external decryption oracle. Note that by the rules in $\mathtt{Game}_2$ the adversary will not be able to force B_2 to perform a decryption query which is disallowed in the IBE security game. All queries associating c_ℓ^* with X_ℓ are immediately answered with $\perp$.
- When A returns a bit b', B_2 will return 1 if $b = b'$ and 0 otherwise.

First we look at the probability that B_2 does not fail. We call the event that B_2 fails $\mathtt{Fail}$. Since the events F_1 and F_2 are independent, we have:

$$\Pr[\neg\mathtt{Fail}] = \Pr[\neg F_1 \wedge \neg F_2] = \Pr[\neg F_1] \cdot [\neg F_2] \geq \frac{1}{mn}.$$

The latter inequality follows from the following observations:

- There is a 1-in-m probability that A will output a policy in which the ℓ-th share must be extracted under the authority corresponding to the external IBE game.
- For any non-trivial policy, there is at least one credential which A cannot extract. Hence there is at least a 1-in-n chance that this is the ℓ-th share.

Let $\hat{b}$ be the bit returned by B_2 and $\bar{b}$ be the secret bit in the external IBE security game. We have

$$\begin{aligned} 2\cdot \text{ Adv }_{\mathtt{IBE}}^{\mathtt{IND-CCA2}}(B_2) &= |\Pr[\hat{b} = 1|\bar{b} = 1] - \Pr[\hat{b} = 1|\bar{b} = 0]| \\ &= \Pr[\neg\mathtt{Fail}] \cdot |\Pr[\hat{b} = 1|\bar{b} = 1 \wedge \neg\mathtt{Fail}] - \Pr[\hat{b} = 1|\bar{b} = 0 \wedge \neg\mathtt{Fail}]|. \end{aligned}$$

Let us now focus on executions of B_2 which do not abnormally terminate. It is clear that algorithm B_2 runs A in the environment of $\mathtt{Game}_2$ if $\ell = n$ and the external encapsulation challenge uses the correct key. Conversely, B_2 runs A in the environment of $\mathtt{Game}_3$ if $\ell = 1$ and the external encapsulation challenge uses the incorrect key. This is true regardless of the fact that the adversary will be able to open up some of the c_j components in the challenge which may contain shares associated with $\mathtt{k}_2$. This is guaranteed by the information-theoretical security of the secret sharing algorithm and by the fact that the adversary is never allowed to obtain a qualifying set of credentials.

Hence, we have

$$\Pr[S_2] = \Pr[\hat{b} = 1 | \ell = n \wedge \bar{b} = 1 \wedge \neg \mathtt{Fail}]$$

and

$$\Pr[S_3] = \Pr[\hat{b} = 1 | \ell = 1 \wedge \bar{b} = 0 \wedge \neg \mathtt{Fail}].$$

Because B_2 generates ℓ uniformly at random at the beginning of its operation, the following summations hold for any execution of B_2:

$$\Pr[\hat{b} = 1 | \bar{b} = 1 \wedge \neg \mathtt{Fail}] = \frac{1}{n} \sum_{i=1}^{n} (\Pr[\hat{b} = 1 | \ell = i \wedge \bar{b} = 1 \wedge \neg \mathtt{Fail}])$$

$$\Pr[\hat{b} = 1 | \bar{b} = 0 \wedge \neg \mathtt{Fail}] = \frac{1}{n} \sum_{i=1}^{n} (\Pr[\hat{b} = 1 | \ell = i \wedge \bar{b} = 0 \wedge \neg \mathtt{Fail}]).$$

Now we observe that for $2 \leq z \leq n$, by construction, B_2 guarantees the following

$$\Pr[\hat{b} = 1 | \ell = z \wedge \bar{b} = 0 \wedge \neg \mathtt{Fail}] = \Pr[\hat{b} = 1 | \ell = (z-1) \wedge \bar{b} = 1 \wedge \neg \mathtt{Fail}].$$

Cancelling out the summation terms, we obtain

$$\begin{aligned} &\Pr[\hat{b} = 1 | \bar{b} = 1 \wedge \neg \mathtt{Fail}] - \Pr[\hat{b} = 1 | \bar{b} = 0 \wedge \neg \mathtt{Fail}] = \\ &\quad = \frac{1}{n} (\Pr[\hat{b} = 1 | \ell = n \wedge \bar{b} = 1 \wedge \neg \mathtt{Fail}] - \Pr[\hat{b} = 1 | \ell = 1 \wedge \bar{b} = 0 \wedge \neg \mathtt{Fail}]). \end{aligned}$$

Putting these results together, we have

$$\mathrm{Adv}_{\mathtt{IBE}}^{\mathtt{IND-CCA2}}(B_2) = \frac{1}{2n} \cdot \Pr[\neg \mathtt{Fail}] \cdot |\Pr[S_2] - \Pr[S_3]|,$$

and finally

$$|\Pr[S_2] - \Pr[S_3]| \leq 2mn^2 \cdot \mathrm{Adv}_{\mathtt{IBE}}^{\mathtt{IND-CCA2}}(B_2),$$

which demonstrates that the advantage of any adversary in $\mathtt{Game}_3$ must be negligibly different from that in $\mathtt{Game}_2$ if the underlying IBE is IND-CCA2 secure.

To complete the proof, we introduce a final game $\mathtt{Game}_4$ where the only difference to $\mathtt{Game}_3$ is the fact that we replace the $\mathtt{aux}_1^*$ component in the challenge by $\mathtt{aux}_2^*$. Again, the information-theoretical security of the secret sharing scheme guarantees that A's view in the two games is identical:

$$\Pr[S_3] = \Pr[S_4].$$

Finally, we have that because no information regarding the secret bit chosen by the challenger can be leaked by the challenge in $\mathtt{Game}_4$, the adversary can have no advantage:

$$\Pr[S_4] = 1/2.$$

Putting together the previous results we obtain the expression in Theorem 2.

$$\mathrm{Adv}^{\mathtt{IND-CCCA}}_{\mathtt{WF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2mn^2 \cdot \mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{IBE}}(B_2).$$

□

Appendix D – Proof of Theorem 4

We present the proof in two stages. First we address IND-CCA2 security, and then IND-CCCA security.

Lemma 3. *The construction is (m, n)-IND-CCA2 secure if the underlying PKE is IND-CCA2 secure and the OTS is UF secure. More precisely we have:*

$$\mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{EFWF-KEM}}(A) \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1) + 2\mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{PKE}}(B_2).$$

Proof. We construct this proof using a sequence of four games $\mathtt{Game}_0, \ldots, \mathtt{Game}_3$. Let A be an adversary against the generic workflow construction. We denote by S_i the event that A guesses the challenge bit correctly in $\mathtt{Game}_i$.

Let $\mathtt{Game}_0$ be the original IND-CCA2 attack game. Hence

$$\mathrm{Adv}^{\mathtt{IND-CCA2}}_{\mathtt{EFWF-KEM}}(A) = |\Pr[S_0] - 1/2|.$$

To obtain $\mathtt{Game}_1$ we introduce a single change: all decapsulation queries where the OTS verification key included in the challenge is reused by the adversary are answered immediately with $\perp$.

We claim that A's probability of success changes negligibly. Let E denote the event that the adversary submits for decapsulation a valid ciphertext $(\mathbf{c}||\sigma)$, different from the challenge ciphertext $(\mathbf{c}^*||\sigma^*)$.

Given that $\mathtt{Game}_0$ and $\mathtt{Game}_1$ are identical, unless E occurs, we have

$$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[E].$$

To show that this difference is negligible, it suffices to demonstrate that $\Pr[E]$ must be negligible. This follows easily from the observation that any adversary that causes E to occur with non-negligible probability can be used to directly construct an algorithm B_1 which wins the UF game against the OTS scheme with advantage $\Pr[E]$. Therefore:

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathrm{Adv}^{\mathtt{UF}}_{\mathtt{OTS}}(B_1).$$

Now we change $\mathtt{Game}_1$ so that decapsulation queries immediately return $\perp$ for all ciphertexts which reuse one or more challenge components c_i^* and corresponding $\mathcal{P}^*$ terms. We call this new game $\mathtt{Game}_2$ and we claim that

$$\Pr[S_1] = \Pr[S_2].$$

To prove this claim, we argue that $\perp$ is the correct decapsulation result for this type of ciphertext in $\mathtt{Game}_1$. To see this, note that all queries which reuse $\mathtt{vk}^*$ are already returning $\perp$, so we only need to consider the case where $\mathtt{vk} \neq \mathtt{vk}^*$. However, by construction, all c_i^* components will return $\mathtt{vk}^*$ when decrypted, causing the decapsulation consistency check to fail.

Finally, we obtain $\mathtt{Game}_3$ by changing the way in which the challenge encapsulation is constructed: the PKE challenge component is constructed using a completely random string of the correct size.

Once again, we claim that A's probability of success changes only negligibly when one moves to $\mathtt{Game}_3$. To prove this claim we show that if $|\Pr[S_2] - \Pr[S_3]|$ is non-negligible, then it is possible to build an algorithm B_2 which runs A as a subroutine, interpolates between the two games, and has non-negligible advantage in the IND-CCA2 game that defines the security of the PKE scheme. B_2 works as follows:

- B_2 generates all the IBE-related parameters itself and obtains the PKE public key from the IND-CCA2 game. These are all handed over to the adversary.
- Eventually the adversary outputs a challenge policy $\mathcal{P}^*$ and B_2 constructs the challenge as follows:
    - B_2 generates the OTS key pair $(\mathtt{vk}^*, \mathtt{sk}^*)$
    - B_2 generates three secret keys $\mathtt{k}_0$, $\mathtt{k}_1$ and $\mathtt{k}_2$.
    - B_2 passes $\mathtt{k}_2||\mathtt{vk}^*$ and a completely random bit string of the same size to the external IND-CCA2 challenge oracle.
    - B_2 uses $\mathtt{k}_1$ to construct the part of the challenge ciphertext which relies on the secret sharing scheme and the IBE scheme.
    - B_2 calculates the OTS and completes the challenge ciphertext.
    - Now B_2 flips a coin b. If $b = 0$ then B_2 hands over $\mathtt{k}_0$ and the challenge to the adversary. Otherwise, the adversary gets $\mathtt{k}_1 \oplus \mathtt{k}_2$ and the challenge ciphertext.
- Eventually the adversary will output its guess b', and B_2 will return 1 if $b = b'$ and 0 otherwise.
- Throughout its entire operation B_2 is able to answer decapsulation queries correctly according to the rules of $\mathtt{Game}_2$. It takes advantage of its knowledge of all the IBE parameters and the external PKE decryption oracle to achieve that. Again, A is not able to force B_2 into placing an invalid query to this oracle because by the rules of $\mathtt{Game}_2$ any query which includes the PKE challenge ciphertext can be immediately answered with $\perp$.

Let $\hat{b}$ denote the secret bit in the PKE IND-CCA2 game, and $\bar{b}$ denote B_2's guess. It is clear that B_2 will run A in an environment consistent with $\mathtt{Game}_2$ or $\mathtt{Game}_3$, depending on whether the PKE challenge oracle encrypts $\mathtt{k}_2||\mathtt{vk}^*$ $(\hat{b} = 0)$ or the random string $(\hat{b} = 1)$. Hence we can write:

$$\Pr[S_2] = \Pr[b = b' | \hat{b} = 0] = \Pr[\bar{b} = 1 | \hat{b} = 0],$$

$$\Pr[S_3] = \Pr[b = b' | \hat{b} = 1] = \Pr[\bar{b} = 1 | \hat{b} = 1].$$

However, by definition, we have that

$$\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-CCA2}}(B_2) = \frac{1}{2} | \Pr[\bar{b} = 1 | \hat{b} = 0] - \Pr[\bar{b} = 1 | \hat{b} = 1] |$$

which leads to

$$| \Pr[S_2] - \Pr[S_3] | = 2\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-CCA2}}(B_2).$$

Finally we observe that in $\mathtt{Game}_3$ the adversary can have no advantage, since no information about $\mathtt{k}_2$ is present in the challenge. Hence

$$\Pr[S_3] = 1/2.$$

The lemma follows from the combination of the game transition results.

$$\mathrm{Adv}_{\mathtt{EFWF-KEM}}^{\mathtt{IND-CCA2}}(A) \leq \mathrm{Adv}_{\mathtt{OTS}}^{\mathtt{UF}}(B_1) + 2\mathrm{Adv}_{\mathtt{PKE}}^{\mathtt{IND-CCA2}}(B_2).$$

□

Lemma 4. *The construction is (m, n)-IND-CCCA secure if the underlying IBE is IND-CCA2 secure, the OTS is UF secure, and the secret sharing scheme is information-theoretically secure. More precisely we have:*

$$\mathrm{Adv}_{\mathtt{EFWF-KEM}}^{\mathtt{IND-CCCA}}(A) \leq \mathrm{Adv}_{\mathtt{OTS}}^{\mathtt{UF}}(B_1) + 2mn^2 \cdot \mathrm{Adv}_{\mathtt{IBE}}^{\mathtt{IND-CCA2}}(B_2).$$

Proof. (*Sketch*) This construction is very similar to the WF-KEM construction that is proven secure in Appendix B. The only difference in the ciphertext is the inclusion of an additional component which corresponds to the PKE encapsulation of a random secret.

Furthermore, in the credential security model for an EFWF-KEM, the adversary knows all the parameters for the underlying PKE, including the secret key, so the additional component in the challenge ciphertext is adding no additional security. We need only to show that the public key component also does not degrade the security of the construction. To do this we analyse how the adversary could obtain additional advantage, and show that this is not feasible.

Global parameters The global parameters for the PKE component are completely independent from the remaining global parameters, so the adversary can obtain no additional advantage through them.

Challenge oracle The challenge ciphertext is constructed using an information theoretically secure 2-out-of-2 splitting of the secret key. One of the shares is handed over to the adversary, since it can open up the public-key component. But this provides no information whatsoever about the encapsulated secret, unless the adversary can learn something about the other share.

Decapsulation Oracle Similarly to what happens in the WF-KEM construction, the OTS scheme combined with the non-malleable embedding of $\mathtt{vk}^*$

in the ciphertext components prevents the adversary from being able to reuse any of the challenge ciphertext components to obtain advantage in a decapsulation query. In a simulation scenario, this means that the strategy used in Appendix B to handle decapsulation queries can also be used in this case.

Credential Extraction Oracle Any credential extraction oracle queries that the adversary may perform will provide no more advantage than it would obtain against the WF-KEM construction, as the IBE component is completely independent of the PKE component.

To summarise, a complete proof of the IND-CCCA security of this EFWF-KEM construction would be almost identical to the proof in Appendix B, the only difference being the fact that the B_1 and B_2 algorithms would need to include the PKE component in their interaction with the adversary. These algorithms would generate the additional parameters themselves, provide them to the adversary, and use them to answer any related decapsulation queries that the adversary might make. □