


# M-Sel: A Message Selection Functional Encryption from Simple Tools

Ahmad Khoureich Ka 

Université Alioune Diop de Bambey, Senegal  
ahmadkhoureich.ka@uadb.edu.sn

**Abstract.** In this paper, we put forward a new practical application of Inner-Product Functional Encryption (IPFE) that we call *Message Selection functional encryption* (M-Sel) which allows users to decrypt selected portions of a ciphertext. In a message selection functional encryption scheme, the plaintext is partitioned into a set of messages  $M = \{m_1, \dots, m_t\}$ . The encryption of  $M$  consists in encrypting each of its elements using distinct encryption keys. A user with a functional decryption key  $sk_{\mathbf{x}}$  derived from a selection vector  $\mathbf{x}$  can access a subset of  $M$  from the encryption thereof and nothing more. Our construction is generic and combines a symmetric encryption scheme and an inner product functional encryption scheme, therefore, its security is tied to theirs. By instantiating our generic construction from a DDH-based IPFE we obtain a message selection FE with constant-size decryption keys suitable for key storage in lightweight devices in the context of Internet of Things (IoT).

**Keywords:** Functional Encryption, Inner-Product Functional Encryption, Adaptive Security.

## 1 Introduction

### 1.1 Functional Encryption.

Unlike traditional Public-Key Encryption ( $\mathcal{PE}$ ), which allows a user with a decryption key to uncover the entire encrypted data, Functional Encryption ( $\mathcal{FE}$ ) allows a finer control over the amount of information accessible to each user from the ciphertext. For a more meaningful formulation, let  $c = \text{Encrypt}(m)$  be a ciphertext and  $sk_f$  a secret key derived from a function  $f$ , the decryption of  $c$  using  $sk_f$  reveals nothing more than  $f(m)$ . The key  $sk_f$  is also called functional decryption key.

Functional encryption first appeared in the forms of Identity-Based Encryption [33,20,15], Searchable Encryption [14,1], Attribute-Based Encryption [32,25,12] and Predicate Encryption [30,27]. But the formal study of functional encryption giving its definitions and various security notions was done later by Boneh, Sahai and Waters [16] and O’Neill [31]. The first  $\mathcal{FE}$  schemes for less general functionality was proposed by Abdalla *et al.* [4]. Their schemes allow

the evaluation of the inner product  $\langle \mathbf{x}, \mathbf{y} \rangle$  of two vectors ( $\mathbf{x}$  encrypted and  $\mathbf{y}$  associated with a decryption key  $sk_{\mathbf{y}}$ ). Therefore, these schemes are called Inner-Product Functional Encryption (IPFE) schemes. The publication of Abdalla *et al.* [4] has aroused a lot of interest among researchers [8,10,7] as application fields are diverse and varied.

Although it is not required that the function associated with the decryption key be hidden, function hiding is very important since it guarantees that sensitive information on the plaintext do not leak. If  $f$  is known, information on the plaintext  $m$  can be gained from  $f(m)$ . Therefore, the inner-product functionality with function hiding is investigated in [13,21,22].

The single input inner-product functionality is extended to the multi-user setting [6,2,18,19,23,29,5]. The latter setting refers to Multi-Input Functional Encryption (MIFE) and Multi-Client Functional Encryption (MCFE). MIFE introduced in [24] is designed for scenarios where input data  $m_1, \dots, m_n$  come from different sources. Each functional decryption key  $sk_f$  is derived from a multi-input function  $f$  that allows computation of  $f(m_1, \dots, m_n)$  from encrypted data  $\text{Encrypt}(m_1), \dots, \text{Encrypt}(m_n)$ . Also, the requirement that nothing beyond  $f(m_1, \dots, m_n)$  is revealed applies. MCFE allows the same computation as MIFE but for input data coming from clients  $1, \dots, n$  who do not trust each other. Each client  $i$  using a secret encryption key generates a ciphertext  $c_i = \text{Encrypt}(m_i, t, i)$  for a plaintext  $m_i$  associated with a tag  $t$  and an index  $i$ . However, MCFE is more restrictive than MIFE on decryption since a decryption key allows the computation of  $f(m_1, \dots, m_n)$  only if the corresponding ciphertexts  $c_1, \dots, c_n$  are labeled with the same tag  $t$ .

## 1.2 This Work.

This work introduces a new type of functional encryption scheme that we call *Message Selection functional encryption* (M-Sel), which has several attractive real-life applications. For example:

**Classified Documents.** The document owner identifies the elements of information that must be classified and establishes the level of classification for each such element. A document  $M = \{m_1, \dots, m_t\} \in 2^{\{0,1\}^*}$  is considered as a set of messages which can be words, phrases, paragraphs, images, etc. To encrypt  $M$ , one computes  $C = \{\text{Encrypt}(sk_1, m_1), \dots, \text{Encrypt}(sk_t, m_t)\}$  where each  $m_{i,i \in [1..t]}$  is encrypted using a secret key  $sk_i$ . Decrypting  $C$  using a functional key  $sk_{\mathbf{x}}$  derived from a selection vector  $\mathbf{x} \in \mathbb{Z}_2^\ell$  yields a subset of  $M$ .

**Image Sharing.** A cloud server hosts images consisting of set of encrypted layers (e.g., map layers in Geographic Information System (GIS)). With their functional decryption key each user accesses a new image obtained by flattening a subset of layers.

**Chat room.** Participants produce encrypted message flows and each of them can only view message flows associated to their functional decryption key.

M-Sel uses a symmetric encryption scheme  $\mathcal{SE}$ , an inner-product functional encryption scheme IPFE and hashing. Our construction can succinctly be presented as follows: the plaintext is partitioned into a set of plaintexts  $M = \{m_1, \dots, m_\ell\} \in 2^{\{0,1\}^*}$ . For each  $m_i \in M$  we pick a random  $s_i \in \mathbb{Z}_p^*$ , derive a bit string  $\sigma_i \leftarrow H(s_i)$  and compute  $u_i = \mathcal{SE}.\text{Encrypt}(\sigma_i, m_i)$ . Then, a vector  $\mathbf{y}_i$  in the canonical basis of  $\mathbb{Z}_2^\ell$  is chosen and  $s_i$  is hidden by computing  $v_i = \text{IPFE}.\text{Encrypt}(mpk, s_i \cdot \mathbf{y}_i)$ . Therefore, the encryption of  $m_i$  is  $(u_i, v_i)$ . A user with a functional decryption key  $sk_{\mathbf{x}}$  derived from a selection vector  $\mathbf{x} \in \mathbb{Z}_2^\ell$  accesses  $m_i$  if  $\text{IPFE}.\text{Decrypt}(sk_{\mathbf{x}}, v_i) = s_i$ . For more details, please refer to the section 3 of the paper.

We prove that our message selection functional encryption scheme have indistinguishable encryptions under a chosen-plaintext attack (IND-CPA) if the underlying  $\mathcal{SE}$  and IPFE schemes are IND-CPA secure.

### 1.3 Related work

Abdalla, Bourse, De Caro and Pointcheval [4] are the first to propose a functional encryption scheme for the inner product functionality. They provided two simple and efficient constructions for IPFE, one based on the Decision Diffie-Hellman assumption (DDH) and the other based on the Learning-With-Errors assumption (LWE). However, the IPFE schemes in [4] are only proven secure in the selective security model where the adversary is asked to declare its challenge messages before the setup of the security game. Subsequently, Agrawal *et al.* [8] proposed an improvement to attain full security under the DDH, LWE, and Decision Composite Residuosity (DCR) assumptions. Chosen ciphertext secure IPFE schemes are first obtained by Benhamouda *et al.* [11]. Their construction is based on projective hash functions with homomorphic properties. These proposals are of great theoretical interest but are not sufficiently efficient for practical applications. Since either they require that the inner product  $\langle \mathbf{x}, \mathbf{y} \rangle$  be small enough for the decryption to work or their parameters sizes are impractical. Finally, Castagnos *et al.* [17] provided IPFE schemes which are efficient for the evaluation of unbounded inner products modulo a prime  $p$ . The efficiency of their constructions is obtained by relying on a cyclic group where the DDH assumption holds containing a subgroup where the discrete logarithm problem is easy.

The message selection functionality can be tackled in the naive way with traditional Hybrid Public-key Encryption  $\mathcal{HP}\mathcal{E}$  which is capable of encrypting arbitrary bit strings. Hybrid public-key encryption combines a symmetric encryption scheme  $\mathcal{SE}$  and a public-key encryption scheme. In this context, the public-key encryption scheme is called key-encapsulation mechanism  $\text{KEM} = (\text{Gen}, \text{Encaps}, \text{Decaps})$ .

The naive scheme is the following. For a plaintext  $M = \{m_1, \dots, m_\ell\} \in 2^{\{0,1\}^*}$ , generate  $\ell$  independent key pairs  $\{sk_i, pk_i\}_{i=1}^\ell$ , set  $mpk = \{pk_1, \dots, pk_\ell\}$  and  $msk = \{sk_1, \dots, sk_\ell\}$ . Apply the encryption algorithm of  $\mathcal{HP}\mathcal{E}$  to  $mpk$  and  $M$  to obtain  $C \leftarrow \{\mathcal{HP}\mathcal{E}.\text{Encrypt}(pk_i, m_i)\}_{i=1}^\ell = \{(c_i, c'_i)\}_{i=1}^\ell$  where  $(c_i, s_i) \leftarrow \text{KEM}.\text{Encaps}(1^\lambda, pk_i)$  and  $c'_i = \mathcal{SE}.\text{Encrypt}(s_i, m_i)$ . A user who wants to access a

subset of  $M$  from  $C$  is given the secret keys  $sk_i$  corresponding to the indices of the selected elements of  $M$ .

In table 1 we summarize the comparison between our approach (M-Sel) based on functional encryption and the naive one based on traditional hybrid public-key encryption in terms of key size and ciphertext size. For this comparison, we consider the instantiation  $\text{M-Sel}_{\text{DDH}}$  of M-Sel from the DDH-based IPFE scheme of [8].  $\text{M-Sel}_{\text{DDH}}$  is described in section 5. Without loss of generality, we assume that the secret keys  $sk_i$  are randomly picked from  $\mathbb{Z}_p$  and the corresponding public keys  $pk_i$  are picked from a cyclic group  $\mathbb{G}$  of prime order  $p$ .

**Table 1.** Comparing M-Sel and the naive approach based on  $\mathcal{HPE}$ .  $|M| = \sum_{i=1}^{\ell} |m_i|$ .

	$mpk$	$msh$	Ciphertext	Decryption key
The naive scheme	$\ell \log p$	$\ell \log p$	$\ell \log p +  M $	$\mathcal{O}(\ell \log p)$
$\text{M-Sel}_{\text{DDH}}$	$\ell \log p$	$2\ell \log p$	$\ell(\ell + 2) \log p +  M $	$2 \log p$

We note that the size of the ciphertext in M-Sel is quadratic in  $\ell$  whereas it is linear in  $\ell$  in the naive solution. However, it is important to note that the advantage of our scheme over the naive one is its short and constant size decryption key which is significantly smaller than that of the naive scheme (which consists of a set of  $\mathcal{O}(\ell)$  secret keys). This makes M-Sel interesting for key storage in lightweight devices in the context of Internet of Things (IoT).

#### 1.4 Organization

The remainder of this paper is organized as follows. Section 2 is devoted to primitives used as components in M-Sel and various settings of encryption. In section 3 we describe the construction of our message selection functional encryption scheme. The security analysis of M-Sel is done in section 4. We show an instantiation of M-Sel from the DDH-based IPFE scheme of [8] in section 5. Section 6 concludes this work.

## 2 Basic Tools

In this section, we recall the syntax of symmetric encryption and of inner-product functional encryption. We also discuss the setting of multi-recipient encryption and the setting of multiple encryptions.

### 2.1 Symmetric Encryption

**Definition 1 (Symmetric Encryption Scheme).** *A symmetric encryption scheme  $\mathcal{SE} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  consists of 3 polynomial-time algorithms:*

1.  $\text{KeyGen}(1^\lambda)$  takes as input a security parameter  $\lambda$  and returns a key  $sk$ .
2.  $\text{Encrypt}(sk, m)$  takes as input a key  $sk$  and a plaintext message  $m \in \{0, 1\}^*$  and returns a ciphertext  $c \leftarrow \text{Encrypt}(sk, m; r) \in \{0, 1\}^*$  where  $r$  is randomly picked from the coins set associated to  $\mathcal{SE}$ . We consider  $r$  to be a part of the ciphertext.
3.  $\text{Decrypt}(sk, c)$  takes as input a key  $sk$  and a ciphertext  $c$  and returns a message  $m$  or an error denoted by the symbol  $\perp$ .

For correctness it is required that  $\text{Decrypt}(sk, \text{Encrypt}(sk, m)) = m$  for all  $m \in \{0, 1\}^*$ .

A symmetric encryption scheme can be used in the multi-recipient setting with randomness re-use. We define multi-recipient encryption as follow:

**Multi-recipient encryption schemes and randomness re-use.** Let  $\mathcal{SE} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  be a standard symmetric encryption scheme. Consider  $n$  receivers, numbered  $1, \dots, n$  each of which has its secret key  $sk_i$ . A sender picks random coins  $r_1, \dots, r_n$  from the coins set associated to  $\mathcal{SE}$  and uses the symmetric encryption scheme  $\overline{\mathcal{SE}} = (\text{KeyGen}, \overline{\text{Encrypt}}, \text{Decrypt})$  to compute  $C \leftarrow \overline{\mathcal{SE}}.\overline{\text{Encrypt}}((sk_1, \dots, sk_n), (m_1, \dots, m_n); (r_1, \dots, r_n)) = (c_1, \dots, c_n)$ , where  $c_i \leftarrow \mathcal{SE}.\text{Encrypt}(sk_i, m_i; r_i)$ . Each receiver  $i$  recovers the plaintext  $m_i = \mathcal{SE}.\text{Decrypt}(sk_i, c_i)$ . The symmetric encryption scheme  $\overline{\mathcal{SE}}$  is termed the Multi-Recipient Encryption Scheme (MRES) associated to  $\mathcal{SE}$ . When all the coins  $r_i$  are equal ( $r_i = r$  for  $i \in [1 .. n]$ ) that is  $c_i \leftarrow \mathcal{SE}.\text{Encrypt}(sk_i, m_i; r)$ ,  $\overline{\mathcal{SE}}$  is termed the Randomness Re-using MRES (RR-MRES) associated to the underlying standard encryption scheme  $\mathcal{SE}$ .

The definition of security for multi-recipient encryption schemes first appeared in [28] and was later refined in [9]. Following [9], we define hereunder indistinguishable encryptions under a chosen-plaintext attack (IND-CPA) experiment for RR-MRES. Let  $\overline{\mathcal{SE}} = (\text{KeyGen}, \overline{\text{Encrypt}}, \text{Decrypt})$  be a randomness re-using symmetric multi-recipient encryption scheme, let  $\mathcal{A}$  be an adversary and  $\lambda$  be the security parameter.  $\mathcal{A}$  has access to an oracle which takes a vector of  $n \in \text{poly}(\lambda)$  messages and outputs a ciphertext vector.

**Experiment**  $\text{Exp}_{\overline{\mathcal{SE}}, \mathcal{A}}^{\text{IND-RR-CPA}}(\lambda)$

$(t, sk_{t+1}, \dots, sk_n) \leftarrow \mathcal{A}(1^\lambda)$  such that  $1 \leq t \leq n \in \text{poly}(\lambda)$   
**For each**  $i \in [1 .. t]$  **do**  $sk_i \leftarrow \overline{\mathcal{SE}}.\text{KeyGen}(1^\lambda)$  **EndFor**  
 $SK \leftarrow (sk_1, \dots, sk_n)$   
 $(m_0^1, \dots, m_0^t; m_1^1, \dots, m_1^t; m_{t+1}, \dots, m_n) \leftarrow \mathcal{A}^{\overline{\mathcal{SE}}(SK, \cdot)}$   
 $b \xleftarrow{R} \{0, 1\}$   
 $M \leftarrow (m_b^1, \dots, m_b^t, m_{t+1}, \dots, m_n)$   
 $r \xleftarrow{R} \text{Coins}$

$C \leftarrow \overline{\mathcal{SE}}.\overline{\text{Encrypt}}(SK, M; r)$   
 $b' \leftarrow \mathcal{A}^{\overline{\mathcal{SE}}(SK, \cdot)}(C)$   
 Return 1 if  $b' = b$ , 0 otherwise

It is mandated that  $|m_0^i| = |m_1^i|$  for all  $i \in [1 .. t]$ . Coins is the coins set associated to  $\mathcal{SE}$ . Notice that when given the security parameter adversary  $\mathcal{A}$  outputs  $n - t$  secret keys and in the challenge phase in addition to messages  $m_0^1, \dots, m_0^t$  and  $m_1^1, \dots, m_1^t$  it provides  $n - t$  other messages. As indicated in [9], this solves the problem of insider attacks ( $\mathcal{A}$  has successfully corrupted  $n - t$  users).

**Definition 2 (IND-CPA security of RR-MRES).** *The advantage of any poly( $\lambda$ )-time adversary  $\mathcal{A}$  in the experiment  $\text{Exp}_{\overline{\mathcal{SE}}, \mathcal{A}}^{\text{IND-RR-CPA}}(\lambda)$  is defined as follow:*

$$\text{Adv}_{\overline{\mathcal{SE}}, \mathcal{A}}^{\text{IND-RR-CPA}}(\lambda) = 2 \cdot \Pr \left[ \text{Exp}_{\overline{\mathcal{SE}}, \mathcal{A}}^{\text{IND-RR-CPA}}(\lambda) = 1 \right] - 1.$$

A randomness re-using symmetric multi-recipient encryption scheme  $\overline{\mathcal{SE}}$  is IND-CPA secure, if the function  $\text{Adv}_{\overline{\mathcal{SE}}, \mathcal{A}}^{\text{IND-RR-CPA}}(\cdot)$  is negligible.

**Theorem 1 (RR-MRES security [9]).** *Fix a symmetric-key encryption scheme  $\mathcal{SE} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  and a polynomial  $n$ . Let  $\overline{\mathcal{SE}} = (\text{KeyGen}, \overline{\text{Encrypt}}, \text{Decrypt})$  be the corresponding RR-MRES. If  $\mathcal{SE}$  is reproducible then for any polynomial-time adversary  $\mathcal{B}$ , there exists a polynomial-time adversary  $\mathcal{A}$ , such that:*

$$\text{Adv}_{\overline{\mathcal{SE}}, \mathcal{B}}^{\text{IND-RR-CPA}}(\lambda) \leq n(\lambda) \cdot \text{Adv}_{\mathcal{SE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$$

Also, [9] states that if  $\mathcal{F}$  is a pseudorandom function family then the symmetric encryption scheme  $\text{CBC}[\mathcal{F}]$  that operates in CBC mode is reproducible. For the remainder of the paper, we consider  $\mathcal{SE}$  to be a symmetric encryption scheme that operates in CBC mode.

## 2.2 Functional Encryption

Functional Encryption is formalized by Boneh, Sahai and Waters in [16]. It is related to the notion of functionality. Inner product functional encryption [4] is a special case of functional encryption and was first provided by Abdalla, Bourse, De Caro and Pointcheval.

**Definition 3 (Functionality).** *A functionality  $F$  defined over  $(\mathcal{K}, \mathcal{M})$  is a function  $F : \mathcal{K} \times \mathcal{M} \rightarrow \Sigma \cup \{\perp\}$ , where  $\mathcal{K}$  is a key space,  $\mathcal{M}$  is a message space and  $\Sigma$  is an output space.*

**Definition 4 (Inner-Product Functional Encryption).** *Inner-product functional encryption is designed for the functionality  $F : \mathcal{R}^\ell \times \mathcal{R}^\ell \rightarrow \mathcal{R} \cup \{\perp\}$  such that  $F(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$  for some ring  $\mathcal{R}$  and a natural number  $\ell$ . An inner product functional encryption scheme IPFE = (Setup, KeyDer, Encrypt, Decrypt) consists of 4 polynomial-time algorithms:*

1.  $\text{Setup}(1^\lambda, 1^\ell)$  takes as input a security parameter  $\lambda$  and a functionality parameter  $\ell$  and returns a master public key  $mpk$  and a master secret key  $msk$ .
2.  $\text{KeyDer}(msk, \mathbf{x})$  takes as input the master secret key  $msk$  and a key  $\mathbf{x} \in \mathcal{R}^\ell$  and derives a secret key  $sk_{\mathbf{x}}$ .
3.  $\text{Encrypt}(mpk, \mathbf{y})$  takes as input the master public key  $mpk$  and a plaintext  $\mathbf{y} \in \mathcal{R}^\ell$  and returns a ciphertext  $c_{\mathbf{y}}$ .
4.  $\text{Decrypt}(mpk, sk_{\mathbf{x}}, c_{\mathbf{y}})$  takes as input the master public key  $mpk$ , a secret key  $sk_{\mathbf{x}}$  and a ciphertext  $c_{\mathbf{y}}$  and returns  $\langle \mathbf{x}, \mathbf{y} \rangle$ .

For correctness, it is required that for all  $\mathbf{x} \in \mathcal{R}^\ell$  and all  $\mathbf{y} \in \mathcal{R}^\ell$ , we have  $\text{Decrypt}(mpk, sk_{\mathbf{x}}, \text{Encrypt}(mpk, \mathbf{y})) = \langle \mathbf{x}, \mathbf{y} \rangle$  or  $\perp$  with negligible probability.

The ring  $\mathcal{R}$  is either  $\mathbb{Z}$  or  $\mathbb{Z}_p$  for some prime number  $p$ . When the inner product is computed in  $\mathbb{Z}_p$ , the  $\text{KeyDer}$  algorithm must monitor secret key requests to avoid giving an adversary decryption keys associated with linearly independent vectors. Indeed, an adversary can request secret keys associated to vectors which are linearly dependent in  $\mathbb{Z}_p$  but linearly independent in  $\mathbb{Z}$ . Such linearly independent secret keys can lead to a solvable system of linear equations where the unknowns are the components of the master secret key. Therefore, the  $\text{KeyDer}$  algorithm must be stateful [8]. Meaning that the adversary obtains redundant information when trying to collect more than  $\ell - 1$  linearly independent secret keys since the  $\text{KeyDer}$  algorithm will return a linear combination of the previous secret keys.

The definition of security for IPFE in the sense of indistinguishable encryptions under a chosen-plaintext attack (IND-CPA) is given via the following experiment. Let  $\mathcal{A}$  be an adversary.

**Experiment**  $\text{Exp}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$

```

Let  $1 \leq q_1 \leq q \in \text{poly}(\lambda)$ ;  $\ell \in \text{poly}(\lambda)$ ;  $S \leftarrow \emptyset$ ;  $S_x \leftarrow \emptyset$ 
 $(mpk, msk) \leftarrow \text{Setup}(1^\lambda, 1^\ell)$ 
For each  $i \in [1 .. q_1]$  do
   $\mathbf{x}_i \leftarrow \mathcal{A}(mpk, S)$ 
   $sk_{\mathbf{x}_i} \leftarrow \text{KeyDer}(msk, \mathbf{x}_i)$ 
   $S \leftarrow S \cup sk_{\mathbf{x}_i}$ 
   $S_x \leftarrow S_x \cup \mathbf{x}_i$ 
EndFor
 $(\mathbf{y}_0, \mathbf{y}_1) \leftarrow \mathcal{A}(mpk, S)$ 
 $b \xleftarrow{R} \{0, 1\}$ 
 $C \leftarrow \text{Encrypt}(mpk, \mathbf{y}_b)$ 

```

$\triangleright$  First phase of secret key queries,  $\mathbf{x}_i \in \mathcal{R}^\ell$ .  
 $\triangleright$  Challenge phase.

For each  $i \in [q_1 \dots q]$  do  
 $\mathbf{x}_i \leftarrow \mathcal{A}(mpk, S, C)$   
 $sk_{\mathbf{x}_i} \leftarrow \text{KeyDer}(msk, \mathbf{x}_i)$   
 $S \leftarrow S \cup sk_{\mathbf{x}_i}$   
 $S_{\mathbf{x}} \leftarrow S_{\mathbf{x}} \cup \mathbf{x}_i$   
 EndFor  
 $b' \leftarrow \mathcal{A}(mpk, S, C)$   
 Return 1 if  $b' = b$ , 0 otherwise

$\triangleright$  Second phase of secret key queries.

It is mandated in the challenge phase and in the second phase of secret key queries that  $\langle \mathbf{x}_i, \mathbf{y}_0 \rangle = \langle \mathbf{x}_i, \mathbf{y}_1 \rangle$  for all  $\mathbf{x}_i \in S_{\mathbf{x}}$ .

**Definition 5 (IND-CPA security of IPFE).** *The advantage of any poly( $\lambda$ )-time adversary  $\mathcal{A}$  in the experiment  $\text{Exp}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$  is defined as follow:*

$$\text{Adv}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) = 2 \cdot \Pr \left[ \text{Exp}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) = 1 \right] - 1.$$

*An inner-product functional encryption scheme IPFE has indistinguishable encryptions under a chosen-plaintext attack, if the function  $\text{Adv}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\cdot)$  is negligible.*

**Multiple Encryptions.** Using the same master public key to encrypt multiple messages is termed Multiple Encryptions (ME). The security of ME is related to that of the based encryption scheme. Hereunder, we define indistinguishable encryptions under a chosen-plaintext attack (IND-CPA) experiment for multiple encryptions. Let  $\mathcal{FE} = (\text{Setup}, \text{KeyDer}, \text{Encrypt}, \text{Decrypt})$  be a functional encryption scheme for the functionality  $F$ , let  $\mathcal{K}$  be the key space, let  $\mathcal{M}$  be the message space, let  $\mathcal{A}$  be an adversary and  $\lambda$  be the security parameter.

**Experiment  $\text{Exp}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\lambda)$**

Let  $1 \leq q_1 \leq q \in \text{poly}(\lambda)$ ;  $t \in \text{poly}(\lambda)$ ;  $S \leftarrow \emptyset$ ;  $S_k \leftarrow \emptyset$   
 $(mpk, msk) \leftarrow \text{Setup}(1^\lambda)$

First phase of secret key queries  $\triangleright$  Syntactically identical to that of  $\text{Exp}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$ .

$(m_0^1, \dots, m_0^t; m_1^1, \dots, m_1^t) \leftarrow \mathcal{A}(mpk, S)$   $\triangleright$  Challenge phase.  
 $b \xleftarrow{R} \{0, 1\}$   
 $C \leftarrow (\text{Encrypt}(mpk, m_0^1), \dots, \text{Encrypt}(mpk, m_0^t))$

Second phase of secret key queries  $\triangleright$  Syntactically identical to that of  $\text{Exp}_{\text{IPFE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$ .

$b' \leftarrow \mathcal{A}(mpk, S, C)$   
 Return 1 if  $b' = b$ , 0 otherwise



It is mandated in the challenge phase and in the second phase of secret key queries that  $F(k_i, m_0^j) = F(k_i, m_1^j)$  for all  $k_i \in S_k \subset \mathcal{K}$  and  $|m_0^j| = |m_1^j|$ ,  $m_0^j, m_1^j \in \mathcal{M}$  for  $j \in [1 .. t]$ .

**Definition 6 (IND-CPA security of ME).** *The advantage of any poly( $\lambda$ )-time adversary  $\mathcal{A}$  in the experiment  $\text{Exp}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\lambda)$  is defined as follow:*

$$\text{Adv}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\lambda) = 2 \cdot \Pr \left[ \text{Exp}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\lambda) = 1 \right] - 1.$$

A functional encryption scheme  $\mathcal{FE}$  has indistinguishable multiple encryptions under a chosen-plaintext attack, if the function  $\text{Adv}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\cdot)$  is negligible.

**Theorem 2 (Multiple encryptions security [26]).** *If a public-key encryption scheme  $\mathcal{PE}$  is CPA-secure, then it also has indistinguishable multiple encryptions.*

### 3 Our Message Selection FE scheme

In this section, we describe our functional encryption scheme for the message selection functionality.

**Definition 7 (Message Selection Functionality).** *Let  $\mathcal{S}$  be the set containing finite sets of messages such that for every  $M \in \mathcal{S}$ ,  $2^M \subset \mathcal{S}$ . Consider  $2^M = \{M_w\}_{w \in \{0,1\}^{|M|}}$  as an indexed family of sets. The message selection functionality is the function  $F : \{0,1\}^n \times \mathcal{S} \rightarrow \mathcal{S} \cup \{\perp\}$  such that  $F(w, M) = M_w$  where  $n$  is a natural number.*

Let  $\mathcal{SE} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  be a symmetric encryption scheme operating in CBC mode with key length  $\kappa$ . Let  $\text{IPFE} = (\text{Setup}, \text{KeyDer}, \text{Encrypt}, \text{Decrypt})$  be an inner-product functional encryption scheme. The construction of M-Sel is as follow:

**Setup**( $1^\lambda, 1^\ell$ ). This algorithm performs the following steps:

1. Choose a cryptographic hash function  $H : \mathbb{Z}_p \rightarrow \{0,1\}^\kappa$  for some prime number  $p > 2^\lambda$ .
2. Call  $\text{IPFE.Setup}(1^\lambda, 1^\ell)$  to obtain a master secret key  $msk$  and a master public key  $mpk$ .

**KeyDer**( $msk, \mathbf{x}$ ) derives from  $\mathbf{x} \in \mathbb{Z}_2^\ell$  a functional key  $sk_{\mathbf{x}} \leftarrow \text{IPFE.KeyDer}(msk, \mathbf{x})$ .

**Encrypt**( $mpk, M$ ). To encrypt a plaintext  $M = \{m_1, m_2, \dots, m_t\} \in 2^{\{0,1\}^*}$  this algorithm performs the following steps:

1. Let  $S \leftarrow \emptyset$ .
2. Let  $\mathbb{B}_\ell$  be the canonical basis of  $\mathbb{Z}_2^\ell$ .
3. Pick a random coins  $r$  from the coins set associated to  $\mathcal{SE}$ .

4. For each  $i \in [1 .. t]$  do:
  - 4.1. Pick a random number  $s_i \in \mathbb{Z}_p^* \setminus S$ ;  $S \leftarrow S \cup s_i$ .
  - 4.2. Compute  $u_i \leftarrow \mathcal{SE}.\text{Encrypt}(H(s_i), m_i; r)$ .
  - 4.3. Choose  $\mathbf{y}_i \in \mathbb{B}_\ell$  and compute  $v_i \leftarrow \text{IPFE}.\text{Encrypt}(mpk, s_i \cdot \mathbf{y}_i)$ .
5. Return the ciphertext  $C = (r, u_1, v_1, \dots, u_t, v_t)$ .

**Decrypt**( $sk_{\mathbf{x}}, C$ ). Using  $sk_{\mathbf{x}}$  to decrypt the ciphertext  $C = (r, u_1, v_1, \dots, u_t, v_t)$ , this algorithm performs the following steps:

1. Let  $M_{\mathbf{x}} = \emptyset$ .
2. For each  $i \in [1 .. t]$  do:
  - 2.1. Compute  $\rho_i = \text{IPFE}.\text{Decrypt}(sk_{\mathbf{x}}, v_i)$ . Note that  $\rho_i = \langle \mathbf{x}, s_i \cdot \mathbf{y}_i \rangle$  is either equal to 0 or  $s_i$ .
  - 2.2. If  $\rho_i = 0$  then return to step 2.1. for the next value of  $i$ . Otherwise, compute  $m_i = \mathcal{SE}.\text{Decrypt}(H(\rho_i), u_i; r)$ .
  - 2.3. If  $m_i = \perp$  then return to step 2.1. for the next value of  $i$ . Otherwise, Set  $M_{\mathbf{x}} \leftarrow M_{\mathbf{x}} \cup m_i$
3. Return the plaintext  $M_{\mathbf{x}}$ .

**Correctness.** For each  $i \in [1 .. t]$ , we have that

$$\begin{aligned} \mathcal{SE}.\text{Decrypt}\left[H\left[\text{IPFE}.\text{Decrypt}(sk_{\mathbf{x}}, v_i)\right], u_i; r\right] &= \mathcal{SE}.\text{Decrypt}\left[H\left[\langle \mathbf{x}, s_i \cdot \mathbf{y}_i \rangle\right], u_i; r\right] \\ &= \begin{cases} \mathcal{SE}.\text{Decrypt}(H(0), u_i; r) \\ \text{or} \\ \mathcal{SE}.\text{Decrypt}(H(s_i), u_i; r) \end{cases} \\ &= \begin{cases} \perp \\ \text{or} \\ m_i \end{cases} \end{aligned}$$

Therefore,  $\text{Decrypt}(sk_{\mathbf{x}}, C) \in 2^M$ .

## 4 Security against Chosen-Plaintext Attack

Here, we prove that M-Sel has indistinguishable encryptions under a chosen-plaintext attack assuming the underlying symmetric CBC encryption scheme  $\mathcal{SE}$  and inner product functional encryption scheme IPFE are IND-CPA secure.

The experiment  $\text{Exp}_{\text{M-Sel}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$  by which we define IND-CPA security for M-Sel is syntactically identical to the experiment  $\text{Exp}_{\mathcal{FE}, \mathcal{A}}^{\text{IND-ME-CPA}}(\lambda)$  given in section 2.2 except for some slight differences presented below:

1. The key space is  $\mathbb{Z}_2^\ell$ .
2.  $F$  is the message selection functionality.

3. In the challenge phase, adversary  $\mathcal{A}$  chooses two distinct sets of messages  $M_0 \leftarrow \{m_0^1, \dots, m_0^t\}$ ,  $M_1 \leftarrow \{m_1^1, \dots, m_1^t\} \in 2^{\{0,1\}^*}$  subject to the restriction that,  $F(\mathbf{x}_i, M_0) = F(\mathbf{x}_i, M_1)$  for all  $\mathbf{x}_i \in S_{\mathbf{x}} \subset \mathbb{Z}_2^\ell$ .

**Theorem 3.** *If the underlying  $\mathcal{SE}$  and IPFE schemes are IND-CPA secure, then M-Sel is IND-CPA secure.*

We recall the definition of perfect secrecy.

**Definition 8.** (Perfectly Secret [26]). *An encryption scheme  $\mathcal{E} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  with message space  $\mathcal{M}$  is perfectly secret if for every probability distribution over  $\mathcal{M}$  every message  $m \in \mathcal{M}$ , and every ciphertext  $c \in \mathcal{C}$  for which  $\Pr[C = c] > 0$ :*

$$\Pr[M = m | C = c] = \Pr[M = m].$$

(The requirement that  $\Pr[C = c] > 0$  is a technical one needed to prevent conditioning on a zero-probability event.)

*Proof (of theorem 3).* Let  $\mathcal{A}$  be an IND-CPA adversary that has advantage  $\epsilon(\lambda)$  against M-Sel by making  $q \in \text{poly}(\lambda)$  secret key queries. Since M-Sel uses  $\mathcal{SE}$  and IPFE, we consider the following extreme cases:

- Case 1: the underlying IPFE seems perfectly secret. Therefore, the security of the RR-MRES based on the underlying  $\mathcal{SE}$  reduces to the security of M-Sel. We present an adversary  $\mathcal{B}$  that interacts with  $\mathcal{A}$  to break the  $\mathcal{SE}$ -based RR-MRES.
- Case 2: the underlying  $\mathcal{SE}$  primitive seems perfectly secret. Therefore, the security of the ME based on the underlying IPFE reduces to the security of M-Sel. We present an adversary  $\mathcal{C}$  that interacts with  $\mathcal{A}$  to break the IPFE multiple encryptions.

Let  $F$  be the message selection functionality.

**Case 1.** Let  $\overline{\mathcal{SE}} = (\text{KeyGen}, \overline{\text{Encrypt}}, \text{Decrypt})$  be the RR-MRES associated to  $\mathcal{SE}$ . Adversary  $\mathcal{B}$  challenges the IND-CPA security of  $\overline{\mathcal{SE}}$ . Let  $\lambda$  be the security parameter and  $\ell$  be the functionality parameter of the underlying IPFE. Consider the following interactions between  $\mathcal{B}$  and  $\mathcal{A}$ :

```

(t, sk_{t+1}, \dots, sk_n) \leftarrow \mathcal{B}(1^\lambda) \text{ such that } 1 \leq t \leq n \in \text{poly}(\lambda)
For each j \in [1 .. t] do: sk_j \xleftarrow{R} \overline{\mathcal{SE}}.\text{KeyGen}(1^\lambda) EndFor ❶
SK \leftarrow (sk_1, \dots, sk_n)
(mpk, msk) \leftarrow \text{M-Sel.Setup}(1^\lambda, 1^\ell) ❷
Let 1 \leq q_1 \leq q \in \text{poly}(\lambda); S \leftarrow \emptyset; S_{\mathbf{x}} \leftarrow \emptyset
For each i \in [1 .. q_1] do
  x_i \leftarrow \mathcal{A}(mpk, S)
  sk_{x_i} \leftarrow \text{M-Sel.KeyDer}(msk, x_i)
  S \leftarrow S \cup sk_{x_i}
  S_{\mathbf{x}} \leftarrow S_{\mathbf{x}} \cup x_i
EndFor

```

▷ First phase of secret key queries.  $\mathbf{x}_i \in \mathbb{Z}_2^\ell$ .

$(\{m_0^1, \dots, m_0^t\}, \{m_1^1, \dots, m_1^t\}) \leftarrow \mathcal{A}(mpk, S) \quad \triangleright$  |Challenge phase.  
 $M_0 \leftarrow \{m_0^1, \dots, m_0^t\}$   
 $M_1 \leftarrow \{m_1^1, \dots, m_1^t\}$   
 $(m_0^1, \dots, m_0^t; m_1^1, \dots, m_1^t; m_{t+1}, \dots, m_n) \leftarrow \mathcal{B}^{\overline{\mathcal{SE}}(SK, \cdot)}(M_0, M_1) \quad \textcircled{3}$   
 $b \xleftarrow{R} \{0, 1\}$   
 $M \leftarrow (m_b^1, \dots, m_b^t, m_{t+1}, \dots, m_n)$   
 $r \xleftarrow{R} \mathbb{Z}_p^*$   
 $(r, u_1, \dots, u_t, u_{t+1}, \dots, u_n) \leftarrow \overline{\mathcal{SE}}.\overline{\text{Encrypt}}(SK, M; r) \quad \textcircled{4}$   
 $C \leftarrow (r, u_1, \dots, u_t, u_{t+1}, \dots, u_n)$   
 Let  $\mathbb{B}_\ell$  be the canonical basis of  $\mathbb{Z}_2^\ell$   
**For each**  $j \in [1 .. t]$  **do**  
      $s_j \xleftarrow{R} \mathbb{Z}_p^*$ ;  $\mathbf{y}_j \xleftarrow{R} \mathbb{B}_\ell$   
      $v_j \leftarrow \text{M-Sel.IPFE.Encrypt}(mpk, s_j \cdot \mathbf{y}_j)$   
**EndFor**  
 $(r, u_1, v_1, \dots, u_t, v_t) \leftarrow \mathcal{B}^{\overline{\mathcal{SE}}(SK, \cdot)}(M_0, M_1, C)$   
**For each**  $i \in [q_1 .. q]$  **do**  
      $\mathbf{x}_i \leftarrow \mathcal{A}(mpk, S, (r, u_1, v_1, \dots, u_t, v_t))$   
      $sk_{\mathbf{x}_i} \leftarrow \text{M-Sel.KeyDer}(msk, \mathbf{x}_i)$   
      $S \leftarrow S \cup sk_{\mathbf{x}_i}$   
      $S_{\mathbf{x}} \leftarrow S_{\mathbf{x}} \cup \mathbf{x}_i$   
**EndFor**  
 $b' \leftarrow \mathcal{A}(mpk, S, (r, u_1, v_1, \dots, u_t, v_t))$   
 $b' \leftarrow \mathcal{B}^{\overline{\mathcal{SE}}(SK, \cdot)}(M_0, M_1, C, b')$

$\triangleright$  |Second phase of secret key queries.

- ①** The challenger sets up the IND-RR-CPA security game for  $\mathcal{B}$ .
- ②**  $\mathcal{B}$  sets up the IND-CPA security game of M-Sel and gets ready to answer to secret keys queries from  $\mathcal{A}$ .
- ③**  $\mathcal{B}$  outputs its challenge messages.
- ④** The challenger outputs the challenge ciphertext for  $\mathcal{B}$  who also prepares the challenge ciphertext for  $\mathcal{A}$ .

It is mandated in the challenge phase and in the second phase of secret key queries that  $F(\mathbf{x}_i, M_0) = F(\mathbf{x}_i, M_1)$  for all  $\mathbf{x}_i \in S_{\mathbf{x}}$  and  $M_0, M_1 \in 2^{\{0,1\}^*}$ .

Assuming that the underlying IPFE primitive seems perfectly secret, we see that adversary  $\mathcal{B}$  interacts with  $\mathcal{A}$  as the latter would interact with the challenger during a chosen plaintext attack against M-Sel. Therefore, we have:

$$\text{Adv}_{\overline{\mathcal{SE}}, \mathcal{B}}^{\text{IND-RR-CPA}}(\lambda) = \text{Adv}_{\text{M-Sel}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) \quad (1)$$

**Case 2.** Let IPFE = (Setup, KeyDer, Encrypt, Decrypt) be an inner-product functional encryption scheme. Adversary  $\mathcal{C}$  challenges the IND-CPA multiple encryptions security of IPFE. Let  $\lambda$  be the security parameter and  $\ell$  be the functionality parameter of the underlying IPFE. Consider the following interactions between  $\mathcal{C}$  and  $\mathcal{A}$ :

Let  $1 \leq q_1 \leq q \in \text{poly}(\lambda)$ ;  $t \in \text{poly}(\lambda)$ ;  $S \leftarrow \emptyset$ ;  $S_{\mathbf{x}} \leftarrow \emptyset$   
 $(mpk, msk) \leftarrow \text{IPFE.Setup}(1^\lambda, 1^\ell)$  ❶  
**For each**  $i \in [1 .. q_1]$  **do**  
      $\mathbf{x}_i \leftarrow \mathcal{A}(mpk, S)$   
      $sk_{\mathbf{x}_i} \leftarrow \text{IPFE.KeyDer}(msk, \mathbf{x}_i)$  ▷ First phase of secret  
      $S \leftarrow S \cup sk_{\mathbf{x}_i}$  key queries.  $\mathbf{x}_i \in \mathbb{Z}_2^\ell$ .  
      $S_{\mathbf{x}} \leftarrow S_{\mathbf{x}} \cup \mathbf{x}_i$   
**EndFor**  
 $(\{m_0^1, \dots, m_0^t\}, \{m_1^1, \dots, m_1^t\}) \leftarrow \mathcal{A}(mpk, S)$  ▷ Challenge phase.  
 $M_0 \leftarrow \{m_0^1, \dots, m_0^t\}$   
 $M_1 \leftarrow \{m_1^1, \dots, m_1^t\}$   
 Let  $\mathbb{B}_\ell$  be the canonical basis of  $\mathbb{Z}_2^\ell$  ❷  
 $r \xleftarrow{R} \mathbb{Z}_p^*$   
**For each**  $j \in [1 .. t]$  **do**  
      $\mathbf{y}_j \xleftarrow{R} \mathbb{B}_\ell$   
      $s_0^j \xleftarrow{R} \mathbb{Z}_p^*$ ;  $\mathbf{e}_0^j \leftarrow s_0^j \cdot \mathbf{y}_j$   
      $s_1^j \xleftarrow{R} \mathbb{Z}_p^*$ ;  $\mathbf{e}_1^j \leftarrow s_1^j \cdot \mathbf{y}_j$   
      $u_i \leftarrow \text{M-Sel.SE.Encrypt}(H(s_0^j), m_0^j; r)$   
**EndFor**  
 $(\mathbf{e}_0^1, \dots, \mathbf{e}_0^t; \mathbf{e}_1^1, \dots, \mathbf{e}_1^t) \leftarrow \mathcal{C}(mpk, M_0, M_1)$  ▷  $\langle \mathbf{x}_i, \mathbf{e}_0^i \rangle = \langle \mathbf{x}_i, \mathbf{e}_1^i \rangle$   
for all  $\mathbf{x}_i \in S_{\mathbf{x}}$ .  
 $b \xleftarrow{R} \{0, 1\}$   
 $(v_1, \dots, v_t) \leftarrow (\text{IPFE.Encrypt}(mpk, \mathbf{e}_0^1), \dots, \text{IPFE.Encrypt}(mpk, \mathbf{e}_0^t))$  ❸  
 $C \leftarrow (v_1, \dots, v_t)$   
 $(u_1, v_1, \dots, u_t, v_t) \leftarrow \mathcal{C}(mpk, M_0, M_1, C)$  ❹  
**For each**  $i \in [q_1 .. q]$  **do**  
      $\mathbf{x}_i \leftarrow \mathcal{A}(mpk, S, (u_1, v_1, \dots, u_t, v_t))$   
      $sk_{\mathbf{x}_i} \leftarrow \text{IPFE.KeyDer}(msk, \mathbf{x}_i)$  ▷ Second phase of secret  
      $S \leftarrow S \cup sk_{\mathbf{x}_i}$  key queries.  
      $S_{\mathbf{x}} \leftarrow S_{\mathbf{x}} \cup \mathbf{x}_i$   
**EndFor**  
 $b' \leftarrow \mathcal{A}(mpk, S, (u_1, v_1, \dots, u_t, v_t))$   
 $b' \leftarrow \mathcal{C}(mpk, M_0, M_1, C, b')$

- ❶ The challenger sets up the IND-ME-CPA security game and gets ready to answer to secret keys queries from  $\mathcal{C}$  who itself gets those secret key queries from  $\mathcal{A}$ .
- ❷  $\mathcal{C}$  prepares the challenge ciphertext for  $\mathcal{A}$ .
- ❸ The challenger outputs the challenge ciphertext for  $\mathcal{C}$ .
- ❹  $\mathcal{C}$  outputs the challenge ciphertext for  $\mathcal{A}$ .

It is mandated in the challenge phase and in the second phase of secret key queries that  $F(\mathbf{x}_i, M_0) = F(\mathbf{x}_i, M_1)$  for all  $\mathbf{x}_i \in S_{\mathbf{x}}$  and  $M_0, M_1 \in 2^{\{0,1\}^*}$ .

Assuming that the underlying  $\mathcal{SE}$  primitive seems perfectly secret, we see that adversary  $\mathcal{C}$  interacts with  $\mathcal{A}$  as the latter would interact with the challenger during a chosen plaintext attack against M-Sel. Therefore, we have:

$$\text{Adv}_{\text{IPFE}, \mathcal{C}}^{\text{IND-ME-CPA}}(\lambda) = \text{Adv}_{\text{M-Sel}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) \quad (2)$$

By summing up equations 1 and 2, we obtain:

$$\text{Adv}_{\text{M-Sel}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) = \frac{1}{2} \cdot \text{Adv}_{\overline{\mathcal{SE}}, \mathcal{B}}^{\text{IND-RR-CPA}}(\lambda) + \frac{1}{2} \cdot \text{Adv}_{\text{IPFE}, \mathcal{C}}^{\text{IND-ME-CPA}}(\lambda)$$

From theorem 1, we know that if a symmetric encryption scheme operates in CBC mode and is IND-CPA secure then the corresponding RR-MRES is also IND-CPA secure. Therefore,  $\text{Adv}_{\overline{\mathcal{SE}}, \mathcal{B}}^{\text{IND-RR-CPA}}(\lambda)$  is negligible. From theorem 2, we also have  $\text{Adv}_{\text{IPFE}, \mathcal{C}}^{\text{IND-ME-CPA}}(\lambda)$  is negligible. Thus,  $\text{Adv}_{\text{M-Sel}, \mathcal{A}}^{\text{IND-CPA}}(\lambda)$  is negligible and we conclude that M-Sel is IND-CPA secure.  $\square$

## 5 Instantiation from a DDH-based IPFE scheme

Instantiation of M-Sel from IPFE schemes of [17] which compute efficiently the inner product is straightforward. Therefore, we give here an instantiation from the DDH-based IPFE scheme of [8] for which the inner product is hard to compute. That DDH-based IPFE scheme (see figure 1) can give short ciphertexts and keys using elliptic curves [3].

### Algorithm Setup( $1^\lambda, 1^\ell$ )

1. Choose a cyclic group  $\mathbb{G}$  of prime order  $p > 2^\lambda$  with generators  $g, h$
2.  $\mathbf{s} = (s_1, \dots, s_\ell) \xleftarrow{R} \mathbb{Z}_p^\ell$
3.  $\mathbf{t} = (t_1, \dots, t_\ell) \xleftarrow{R} \mathbb{Z}_p^\ell$
4. For each  $i \in [1 .. \ell]$   
compute  $h_i = g^{s_i} \cdot h^{t_i}$
5. Return  
 $msk = (\mathbf{s}, \mathbf{t})$ ,  
 $mpk = (\mathbb{G}, g, h, \{h_i\}_{i=1}^\ell)$

### Algorithm Encrypt( $mpk, \mathbf{y}$ )

- $$\mathbf{y} = (y_1, \dots, y_\ell) \in \mathbb{Z}_p^\ell$$
1. Pick  $r \xleftarrow{R} \mathbb{Z}_p^*$
  2. Compute  $C = g^r, D = h^r$
  3. Compute  $\{E_i = g^{y_i} \cdot h_i^r\}_{i=1}^\ell$
  4. Return  $C_{\mathbf{y}} = (C, D, E_1, \dots, E_\ell)$

### Algorithm KeyDer( $msk, \mathbf{x}$ )

- $$\mathbf{x} = (x_1, \dots, x_\ell) \in \mathbb{Z}_p^\ell$$
1. Compute  $\alpha = \sum_{i=1}^\ell s_i \cdot x_i$
  2. Compute  $\beta = \sum_{i=1}^\ell t_i \cdot x_i$
  3. Return  $sk_{\mathbf{x}} = (\alpha, \beta)$

### Algorithm Decrypt( $mpk, sk_{\mathbf{x}}, C_{\mathbf{y}}$ )

- $$sk_{\mathbf{x}} = (\alpha, \beta)$$
1. Compute  
 $E_{\mathbf{x}} = (\prod_{i=1}^\ell E_i^{x_i}) / (C^\alpha \cdot D^\beta)$
  2. Return  $\langle \mathbf{x}, \mathbf{y} \rangle = \log_g(E_{\mathbf{x}})$

**Fig. 1.** DDH-based IPFE scheme of Agrawal *et al.* [8].

To suit our M-Sel scheme, we customize the decryption algorithm so that it does not compute the actual value of  $\langle \mathbf{x}, \mathbf{y} \rangle$  but returns  $E_{\mathbf{x}} = g^{\langle \mathbf{x}, \mathbf{y} \rangle}$ . In the remainder of the paper, to avoid confusion, we denote  $\text{Decrypt}^*$  this customization of  $\text{Decrypt}$ . This immunizes M-Sel against the main drawback of DDH-based IPFE schemes that is the inner product  $\langle \mathbf{x}, \mathbf{y} \rangle$  must be small enough for the decryption to work.

**DDH-based M-Sel.** In the description hereunder of  $\text{M-Sel}_{\text{DDH}}$ , we only show steps in our generic construction (in section 3) that change.

Let  $\mathcal{SE} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$  be a symmetric encryption scheme operating in CBC mode with key length  $\kappa$ . Let  $\text{IPFE}_{\text{DDH}} = (\text{Setup}, \text{KeyDer}, \text{Encrypt}, \text{Decrypt})$  be the DDH-based IPFE scheme of [8]. Let  $\mathbb{G}$  be a cyclic group of prime order  $p > 2^\lambda$  with generators  $g, h$ .

**Setup**( $1^\lambda, 1^\ell$ ).

1. Choose a cryptographic hash function  $H : \mathbb{G} \rightarrow \{0, 1\}^\kappa$ .

**KeyDer**( $msk, \mathbf{x}$ ). *No changes.*

**Encrypt**( $mpk, M$ ).  $M = \{m_1, m_2, \dots, m_t\} \in 2^{\{0,1\}^*}$ .

- 4.2. Compute  $u_i \leftarrow \mathcal{SE}.\text{Encrypt}(H(g^{s_i}), m_i; r)$ .

**Decrypt**( $sk_{\mathbf{x}}, C$ ).  $C = (r, u_1, v_1, \dots, u_t, v_t)$ .

- 2.1. Compute  $\rho_i = \text{IPFE}_{\text{DDH}}.\text{Decrypt}^*(sk_{\mathbf{x}}, v_i)$ . Note that  $\rho_i = g^{\langle \mathbf{x}, s_i \cdot \mathbf{y}_i \rangle}$  is either equal to the identity element  $1_{\mathbb{G}}$  or  $g^{s_i}$ .
- 2.2. If  $\rho_i = 1_{\mathbb{G}}$  then return to step 2.1. for the next value of  $i$ . Otherwise, compute  $m_i = \mathcal{SE}.\text{Decrypt}(H(\rho_i), u_i; r)$ .

## 6 Conclusion

We proposed a generic construction for the message selection functionality called M-Sel that achieves security against adaptive adversaries. M-Sel can be efficient and practical when instantiated with an efficient inner-product functional encryption (IPFE) scheme. An instantiation of M-Sel from a DDH-based IPFE was also presented. The latter instantiation has short and constant size decryption key thus, suitable for key storage in lightweight devices in the context of Internet of Things (IoT).

**Acknowledgements.** We would like to thank the anonymous reviewers for providing helpful comments and suggestions about this work.

## References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In: Shoup, V. (ed.) *Advances in Cryptology – CRYPTO 2005*. pp. 205–222. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
2. Abdalla, M., Benhamouda, F., Gay, R.: From single-input to multi-client inner-product functional encryption. In: Galbraith, S.D., Moriai, S. (eds.) *Advances in Cryptology – ASIACRYPT 2019*. pp. 552–582. Springer International Publishing, Cham (2019)
3. Abdalla, M., Bourse, F., Caro, A.D., Pointcheval, D.: Better security for functional encryption for inner product evaluations. *Cryptology ePrint Archive*, Paper 2016/011 (2016), <https://eprint.iacr.org/2016/011>
4. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Katz, J. (ed.) *Public-Key Cryptography – PKC 2015*. pp. 733–751. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
5. Abdalla, M., Bourse, F., Marival, H., Pointcheval, D., Soleimani, A., Waldner, H.: Multi-client inner-product functional encryption in the random-oracle model. In: Galdi, C., Kolesnikov, V. (eds.) *Security and Cryptography for Networks*. pp. 525–545. Springer International Publishing, Cham (2020)
6. Abdalla, M., Catalano, D., Fiore, D., Gay, R., Ursu, B.: Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In: Shacham, H., Boldyreva, A. (eds.) *Advances in Cryptology – CRYPTO 2018*. pp. 597–627. Springer International Publishing, Cham (2018)
7. Abdalla, M., Catalano, D., Gay, R., Ursu, B.: Inner-product functional encryption with fine-grained access control. In: Moriai, S., Wang, H. (eds.) *Advances in Cryptology – ASIACRYPT 2020*. pp. 467–497. Springer International Publishing, Cham (2020)
8. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) *Advances in Cryptology – CRYPTO 2016*. pp. 333–362. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
9. Bellare, M., Boldyreva, A., Staddon, J.: Randomness re-use in multi-recipient encryption schemes. In: Desmedt, Y.G. (ed.) *Public Key Cryptography — PKC 2003*. pp. 85–99. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
10. Benhamouda, F., Bourse, F., Lipmaa, H.: Cca-secure inner-product functional encryption from projective hash functions. In: Fehr, S. (ed.) *Public-Key Cryptography – PKC 2017*. pp. 36–66. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)
11. Benhamouda, F., Bourse, F., Lipmaa, H.: Cca-secure inner-product functional encryption from projective hash functions. In: Fehr, S. (ed.) *Public-Key Cryptography – PKC 2017*. pp. 36–66. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)
12. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07). pp. 321–334 (2007). <https://doi.org/10.1109/SP.2007.11>
13. Bishop, A., Jain, A., Kowalczyk, L.: Function-hiding inner product encryption. In: Iwata, T., Cheon, J.H. (eds.) *Advances in Cryptology – ASIACRYPT 2015*. pp. 470–491. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
14. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) *Advances in Cryptology*



- EUROCRYPT 2004. pp. 506–522. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
15. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) *Advances in Cryptology — CRYPTO 2001*. pp. 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
  16. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) *Theory of Cryptography*. pp. 253–273. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
  17. Castagnos, G., Laguillaumie, F., Tucker, I.: Practical fully secure unrestricted inner product functional encryption modulo  $p$ . In: Peyrin, T., Galbraith, S.D. (eds.) *Advances in Cryptology - ASIACRYPT 2018*. *Lecture Notes in Computer Science*, vol. 11273, pp. 733–764. Springer (2018). [https://doi.org/10.1007/978-3-030-03329-3\\_25](https://doi.org/10.1007/978-3-030-03329-3_25)
  18. Chotard, J., Dufour Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Decentralized multi-client functional encryption for inner product. In: Peyrin, T., Galbraith, S. (eds.) *Advances in Cryptology – ASIACRYPT 2018*. pp. 703–732. Springer International Publishing, Cham (2018)
  19. Chotard, J., Dufour-Sans, E., Gay, R., Phan, D.H., Pointcheval, D.: Multi-client functional encryption with repetition for inner product. *Cryptology ePrint Archive*, Paper 2018/1021 (2018), <https://eprint.iacr.org/2018/1021>
  20. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) *Cryptography and Coding*. pp. 360–363. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
  21. Datta, P., Dutta, R., Mukhopadhyay, S.: Functional encryption for inner product with full function privacy. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) *Public-Key Cryptography – PKC 2016*. pp. 164–195. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
  22. Datta, P., Dutta, R., Mukhopadhyay, S.: Strongly full-hiding inner product encryption. *Theoretical Computer Science* **667**, 16–50 (2017), <https://doi.org/10.1016/j.tcs.2016.12.024>
  23. Datta, P., Okamoto, T., Tomida, J.: Full-hiding (unbounded) multi-input inner product functional encryption from the  $k$ -linear assumption. In: Abdalla, M., Dahab, R. (eds.) *Public-Key Cryptography – PKC 2018*. pp. 245–277. Springer International Publishing, Cham (2018)
  24. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Nguyen, P.Q., Oswald, E. (eds.) *Advances in Cryptology – EUROCRYPT 2014*. pp. 578–602. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
  25. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security*. p. 89–98. CCS '06, Association for Computing Machinery, New York, NY, USA (2006), <https://doi.org/10.1145/1180405.1180418>
  26. Katz, J., Lindell, Y.: *Introduction to Modern Cryptography*, Second Edition. Chapman & Hall/CRC, New York (2014), <https://doi.org/10.1201/b17668>
  27. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) *Advances in Cryptology – EUROCRYPT 2008*. pp. 146–162. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)

28. Kurosawa, K.: Multi-recipient public-key encryption with shortened ciphertext. In: Naccache, D., Paillier, P. (eds.) *Public Key Cryptography*. pp. 48–63. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
29. Nguyen, K., Pointcheval, D., Schädlich, R.: Function-hiding dynamic decentralized functional encryption for inner products. *Cryptology ePrint Archive*, Paper 2022/1532 (2022), <https://eprint.iacr.org/2022/1532>
30. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) *Advances in Cryptology – CRYPTO 2010*. pp. 191–208. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
31. O’Neill, A.: Definitional issues in functional encryption. *Cryptology ePrint Archive*, Paper 2010/556 (2010), <https://eprint.iacr.org/2010/556>
32. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) *Advances in Cryptology – EUROCRYPT 2005*. pp. 457–473. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
33. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) *Advances in Cryptology*. pp. 47–53. Springer Berlin Heidelberg, Berlin, Heidelberg (1985)