Secret Sets in the Public-Key Setting

Abstract

A secret set is a representation of a subset of users of a given universe, satisfying the following properties: any user of the universe can check whether he is or is not a member of the subset; no one can check if another user of the universe is or is not a member; no one can determine the size of the subset. In this paper we focus our attention on secret sets in the public-key setting.

1 Introduction

Secret Sets. Fifteen years ago, Molva and Tsudik [15] put forward the notion of *secret set* as a method to enhance user privacy. Loosely speaking, a secret set is a *representation* of a subset of users of a given universe such that *any* user of the universe can check whether he is or is not a member of the subset, *no one* can check if another user of the universe is or is not a member, and *no one* can determine the size of the subset. The last two properties should hold also against coalitions of users. The authors proposed some constructions and showed how secret sets can be useful to protect receivers' privacy in multicast communications, and against traffic analysis of mobile devices.

Later on, De Santis and Masucci [9] provided a formal treatment of the notion. They defined unconditionally secure secret sets by using the language of information theory, showed lower bounds in terms of needed number of bits on user storage, on representation length of a secret set, and on the randomness needed to set up a scheme, and proved the bounds are tight. Moreover, they defined computationally secure secret sets in terms of *indistinguishability* of representations associated to different sets, and showed that such schemes exists if and only if semantically secure symmetric encryption exists.

Micali et al. in [14] put forward a related notion: the notion of *zero-knowledge set*. A zero-knowledge set is a method through which a prover can construct a representation of a set of strings S of a given universe $\mathcal{U}$ such that, for any string $x \in \mathcal{U}$, he is able to prove non-interactively and in zero-knowledge whether $x \in S$ or $x \notin S$. In particular, the representation of S does not leak any other information about S, e.g., the size of S. The authors showed that zero-knowledge sets exist if the discrete logarithm problem is hard. Several papers have further focused on Micali et al.'s work, e.g., see [7, 5, 6]

A third related notion was introduced in [4]: *private broadcast encryption*. Indeed, in many content distribution systems it is important to both restrict access to content to authorized users and to protect *their identities*. A private broadcast encryption scheme is exactly a mechanism to encrypt a broadcast message such that only authorized users can decrypt the message and read the content and, at the same time, their identities are kept secret, even from each other. Such a notion has been further studied in [13], under the name of *anonymous broadcast encryption*.

The growing need for user privacy has lead to the development of general notions and efficient tools for building privacy-preserving applications. Among them, two encryption notions are frequently used: the notion of *key-privacy* in public-key encryption [2], and the notion of *robust encryption* [1]. They have been used in [4] and, even if they were not formalised at that time, in a certain sense, they were underlying some of the constructions in [15].

Motivations and goals. In this paper, by using the currently available knowledge and tools, developed during the last years, we look again at secret sets, focusing our attention on constructions for secret sets in the public-key setting. Indeed, several issues are still open: the authors of [15] proposed some constructions based on encryption schemes and the Chinese Remainder Theorem and suggested two important applications, but the treatment they provided was quite informal. No proofs were given. On the other hand, [9] provided a formal treatment, but in the computational case the authors looked mainly at the symmetric setting. Moreover, in both papers, no efficient construction hides the size of the set S, which is disclosed to the users.

Our contribution. We propose a *model* for secret sets in the public key setting which formalizes the intuition about the primitive, and we present some *constructions*. For each of them, we show which security requirements set in the model the construction achieves, and under which computational assumptions. The *proofs*, hence, clarify what is really needed to get certain security properties.

2 Encryption: different flavors

In this section we set up the notation and revise some properties an encryption scheme should achieve to be useful in privacy-preserving applications.

Notation. Let k be a security parameter, expressed in unary notation 1^k.

Poly functions. We denote with $poly(k)$ any positive function $f(\cdot)$ upper bounded by a polynomial $p(\cdot)$, i.e., $f(k) \leq p(k)$, for any $k \geq 1$. We use the $poly(k)$ notation any time we wish to skip details and stress that the computations/resources needed are polynomial.

Efficient Algorithms. An efficient algorithm is a probabilistic algorithm running in $poly(k)$ time. For short, efficient algorithms are referred to as *ppt algorithms*.

Negligible functions. A function $f(\cdot)$ is negligible if it vanishes faster than the inverse of any fixed positive polynomial. That is, for any positive integer c, there exists an integer k_0 such that $f(k) \leq \frac{1}{k^c}$, for any $k \geq k_0$. We denote by $negl(k)$ a negligible function.

Probabilistic assignments. If S is a probability space, then $x \leftarrow S$ denotes the act of choosing an element at random according to the probability distribution defined on S. On the other hand, if F is a finite set, then $x \leftarrow F$ denotes the act of choosing x uniformly from F.

Algorithms and probabilistic spaces. If $A(\cdot)$ is a probabilistic algorithm, then, for any x, the notation $A(x)$ refers to the probability space that assigns to the string σ the probability that A, on input x, outputs σ.

Probabilistic experiments. If p is a predicate, and $S_1, \ldots, S_n$ are probability spaces, then the notation $Pr[x_1 \leftarrow S_1; \ldots; x_n \leftarrow S_n : p(x_1, \ldots, x_n)]$ denotes the probability that $p(x_1, \ldots, x_n)$ will be true after the ordered execution of the probabilistic assignments $x_1 \leftarrow S_1; \ldots; x_n \leftarrow S_n$.

Tools. Public-key encryption schemes satisfying special properties are needed in the following section, where secret set constructions are described and analysed. We state here all notions used later on. Let

us start from the basic notion of public-key encryption [12].

Definition 2.1 *A* public-key encryption scheme *is a triple of probabilistic polynomial-time algorithms* $\Pi = (Gen, Enc, Dec)$*, along with a message space* $\mathcal{M}$ *and a ciphertext space* $\mathcal{C}$*, such that:*

1. *The* key-generation algorithm Gen *takes as input the security parameter* 1^k *and outputs a pair of keys* (pk, sk)*, i.e.,* $(pk, sk) \leftarrow Gen(1^k)$*. We refer to the first of these as the* public key *and the second as the* private key*.*

2. *The* encryption algorithm Enc *takes as input a public key* pk *and a message* $m \in \mathcal{M}$ *and outputs a ciphertext* $c \in \mathcal{C}$*, i.e.,* $c \leftarrow Enc_{pk}(m)$*.*

3. *The* decryption algorithm Dec *takes as input a private key* sk *and a ciphertext* c*, and outputs a message* m *or a special symbol* $\perp$ *denoting failure.* Dec *is deterministic, i.e.,* $m = Dec_{sk}(c)$*.*

It is required that there exists a negligible function $negl(k)$ *such that, for any* k*, it holds that*

$$Pr[(pk, sk) \leftarrow Gen(1^k); m \leftarrow \mathcal{M}; c \leftarrow Enc_{pk}(m) : Dec_{sk}(c) \neq m] \leq negl(k).$$

The condition essentially states that the encryption scheme has to be correct: encrypting any message m with pk and, later on, decrypting the ciphertext with sk, gives back the message m. The definition is very general and allows a negligible probability of failure.

Intuitively an encryption scheme is secure if a ciphertext does not leak *any partial* information about the plaintext. Such a requirement has been formalised in [10] under the notion of *semantic security*. However, since it is not easy to deal with it, a second notion, easier to manage, was introduced, and it was shown to be *equivalent* to semantic security. Specifically, an encryption scheme has *indistinguishable encryptions* if, given any two messages m_0 and m_1, and a ciphertext c obtained by encrypting one of them chosen uniformly at random, any ppt algorithm $\mathcal{A}$ is able to associates c to the right message only with negligible probability. The indistinguishability paradigm is the paradigm in use.

Any entity $\mathcal{A}$ which attacks an encryption scheme is referred to as an *adversary* and is modeled as a ppt algorithm. Given a public-key encryption scheme $\Pi = (Gen, Enc, Dec)$ and an adversary $\mathcal{A}$, the security of Π w.r.t. $\mathcal{A}$ can be defined throught an experiment, which clarifies $\mathcal{A}'$s abilities and the properties the scheme should exhibit. Let us consider the following chosen ciphertext security experiment, run by a challenger C :

Experiment 2.2 $Pub^{CCA}_{\Pi,\mathcal{A}}$

1. *C gets* $(pk, sk) \leftarrow Gen(1^k)$*.*

2. $\mathcal{A}$ *receives* pk *from* C *and gets access poly(k) times to a decryption oracle* $Dec_{sk}(\cdot)$*. Then, it outputs a pair of messages* (m_0, m_1) *of the same length.*

3. *C chooses* $b \leftarrow \{0, 1\}$*, and then computes a ciphertext* $c \leftarrow Enc_{pk}(m_b)$*, given to* $\mathcal{A}$*.*

4. $\mathcal{A}$ *keeps interacting with the decryption oracle* $Dec_{sk}(\cdot)$*, but may not request a decryption of* c *itself. Finally,* $\mathcal{A}$ *outputs a bit* b'*.*

5. *If* $b' = b$*, then C outputs* 1*; otherwise, it outputs* 0*.*

Notice that, in the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment, $\mathcal{A}$ has the abilities to encrypt any message it wishes, by using the public key pk, and to decrypt any ciphertext at its choice, apart the challenge ciphertext c, by querying the decryption oracle $Dec_{sk}(\cdot)$. Hence, the experiment formalizes a real-life setting in which a powerful adversary may eavesdrop the communications and get extra knowledge about the encryption scheme, by choosing ciphertexts and receiving the corresponding plaintexts. If after some work, $\mathcal{A}$ is still unable to distinguish an encryption of m_0 from an encryption of m_1, then the scheme Π is secure. More precisely:

Definition 2.3 *A public-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions under chosen-ciphertext attack (for short, it is cca-secure) if, for all probabilistic polynomial time adversaries $\mathcal{A}$, there exists a negligible function $negl(k)$ such that:*

$$Pr[Pub^{CCA}_{\Pi,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

Notice that cca-security is the standard notion required by an encryption scheme in general applications. However, when it is sufficient protection against an adversary which *simply eavesdrops* the communications, the weaker notion of indistinguishability of encryptions under chosen plaintext attacks, for short cpa-security, is enough. It is defined by taking off in experiment $Pub^{CCA}_{\Pi,\mathcal{A}}$ the access to the decryption oracle. The corresponding experiment is referred to as $Pub^{CPA}_{\Pi,\mathcal{A}}$.

An important property a public-key encryption scheme may exhibit, which is helpful in building applications providing user anonymity, is the *key-privacy* property [2]: roughly speaking, an adversary is unable to understand *which* public key has been used to compute a given ciphertext. Notice that, Definition 2.3, formalizes ciphertext indistinguishability but *does not say anything* about partial information a ciphertext might release about the public key with whom it was created. More precisely, the key-privacy property can be defined through the following experiment, run by a challenger C :

Experiment 2.4 $KPriv^{CCA}_{\Pi,\mathcal{A}}$

1. *C generates $(pk_0, sk_0) \leftarrow Gen(1^k)$ and $(pk_1, sk_1) \leftarrow Gen(1^k)$*
2. *$\mathcal{A}$ receives pk_0, pk_1 from C and gets access poly(k) times to the decryption oracles $Dec_{sk_0}(\cdot)$ and $Dec_{sk_1}(\cdot)$. Then, it outputs a message m.*
3. *C chooses $b \leftarrow \{0,1\}$, and then computes a ciphertext $c \leftarrow Enc_{pk_b}(m)$, given to $\mathcal{A}$.*
4. *$\mathcal{A}$ keeps interacting with the decryption oracles $Dec_{sk_0}(\cdot)$ and $Dec_{sk_1}(\cdot)$, but may not request a decryption of c itself. Finally, $\mathcal{A}$ outputs a bit b'.*
5. *If $b' = b$, then C outputs 1; otherwise, it outputs 0.*

Notice that, as in the previous experiment, $\mathcal{A}$ has the ability to acquire knowledge by querying the decryption oracles. The scheme Π satisfies the key-privacy property if $\mathcal{A}$ is still unable at the end of the experiment to distinguish an encryption of the chosen message m with pk_0 from an encryption of m with pk_1. More precisely:

Definition 2.5 *A public-key encryption scheme $\Pi = (Gen, Enc, Dec)$ is key-private under chosen-ciphertext attack (for short, it is ik-cca private) if, for all probabilistic polynomial time adversaries $\mathcal{A}$, there exists a negligible function $negl(k)$ such that:*

$$Pr[KPriv^{CCA}_{\Pi,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

As before, we can consider a weaker notion of adversary by taking off $\mathcal{A}$'s access to the decryption oracle in experiment $KPriv^{CCA}_{\Pi,\mathcal{A}}$. The corresponding experiment is noted by $KPriv^{CPA}_{\Pi,\mathcal{A}}$ and a scheme achieving it is referred to as ik-cpa private. In [2] it was shown that, under the decisional Diffie-Hellman assumption, the ElGamal encryption scheme is ik-cpa private, and (a slightly modified version of) the Cramer-Shoup encryption scheme is ik-cca private.

Another useful property an encryption scheme may enforce is the so called *robustness* [1]: basically, a robust encryption scheme guarantees that a ciphertext c^*, produced by using encryption key pk_0, can only be decrypted by using sk_0. There is *no* other key-pair (pk_1, sk_1) such that $Dec_{sk_1}(c^*) \neq \perp$. Again, notice that Definition 2.1 formalizes correctness of decryption by using the secret key associated to the public key with whom the ciphertext was created (with overwhelming probability), but *does not say anything* about the result of decrypting c^* by using a *different* secret key. Formally, a weak form of robustness is defined through the following experiment $WROB_{\Pi,\mathcal{A}}$, run by a challenger C:

Experiment 2.6 $WROB_{\Pi,\mathcal{A}}$

1. *C generates two key-pairs, i.e.,* $(pk_i, sk_i) \leftarrow Gen(1^k)$, *for* $i = 0, 1$.
2. $\mathcal{A}$ *chooses* $m \in \mathcal{M}$, *computes* $c^* = Enc_{pk_0}(m)$, *and sends* c^* *to C.*
3. *C computes* $m_0 = Dec_{sk_0}(c^*)$ *and* $m_1 = Dec_{sk_1}(c^*)$.
4. *If* $(m_0 \neq \perp \wedge m_1 \neq \perp)$, *then C outputs* 1*; otherwise* 0.

If any adversary has negligible probability of succeeding in the above experiment, then the encryption scheme is weakly robust. More precisely:

Definition 2.7 *A public-key encryption scheme* $\Pi = (Gen, Enc, Dec)$ *is* weakly robust *if, for all probabilistic polynomial time adversaries* $\mathcal{A}$, *there exists a negligible function* $negl(k)$ *such that:*

$$Pr[WROB_{\Pi,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

A stronger notion allows the adversary $\mathcal{A}$ to construct the challenge ciphertext c^* in *any way*, e.g., even by not choosing a message m and using the public key pk_0 in step 2. The corresponding experiment is denoted as $SROB_{\Pi,\mathcal{A}}$ and if the adversary has negligible probability of succeeding, then the encryption scheme is strongly robust. More precisely:

Definition 2.8 *A public-key encryption scheme* $\Pi = (Gen, Enc, Dec)$ *is* strongly robust *if, for all probabilistic polynomial time adversaries* $\mathcal{A}$, *there exists a negligible function* $negl(k)$ *such that:*

$$Pr[SROB_{\Pi,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

In [1] the authors proposed two generic transformations to get weak robustness and strong robustness.

Another fundamental tool in our constructions are digital signatures. In particular, we use existentially unforgeable under chosen-message attacks digital signature schemes. Let us start from the basic definition of digital signature scheme.

Definition 2.9 *A digital* signature scheme *is a triple of probabilistic polynomial-time algorithms* $\Pi = (Gen, Sign, Ver)$ *along with a message space* $\mathcal{M}$, *satisfying the following:*

1. *The* key-generation algorithm Gen *takes as input a security parameter* 1^k *and outputs a pair of keys* (vk, sk), *the* verification key *and the* signing key.

2. *The* signing algorithm $Sign$ *takes as input a signing key* sk *and a message* $m \in \mathcal{M}$. *It outputs a signature* σ, *i.e.,* $\sigma \leftarrow Sign_{sk}(m)$.

3. *The deterministic* verification algorithm Ver *takes as input the verification key* vk, *a message* m *and a signature* σ. *It outputs a bit* b, *with* $b = 1$ *meaning* valid, *and* $b = 0$ *meaning* invalid, *i.e.,* $b = Ver_{vk}(m, \sigma)$

It is required that there exists a negligible function $negl(k)$ *such that, for any* k, *it holds that*

$$Pr[(vk, sk) \leftarrow Gen(1^k); m \leftarrow \mathcal{M}; \sigma \leftarrow Sign_{sk}(m) : Ver_{vk}(m, \sigma) \neq 1] \leq negl(k).$$

The existential unforgeability under a chosen-message attack property, can be defined through the following experiment, run by a challenger C :

Experiment 2.10 $Sig\text{-}forge_{\mathcal{A},\Pi}$

1. *C generates a key-pair i.e.,* $(vk, sk) \leftarrow Gen(1^k)$.
2. $\mathcal{A}$ *gets* vk *and oracle access to* $Sign_{sk}(\cdot)$. *The oracle returns a signature* $Sign_{sk}(m)$ *for any message* m *of the adversary's choice. The oracle can be queried poly(k) times.*
3. $\mathcal{A}$ *outputs a pair* (m, σ), *where* m *was not queried before to the oracle.*
4. *C outputs* 1 *if and only if* $Ver_{vk}(m, \sigma) = 1$; *C outputs* 0 *otherwise.*

Definition 2.11 *A digital signature scheme* $\Pi = (Gen, Sign, Ver)$ *is* existentially unforgeable under a chosen-message attack *if, for all probabilistic polynomial time algorithms* $\mathcal{A}$, *there exists a negligible function* $negl(k)$ *such that*

$$Pr[Sig\text{-}forge_{\mathcal{A},\Pi} = 1] \leq negl(k).$$

3 Secret sets: the model.

In this section we formalize the notion of secret set in the public-key setting, and state the privacy requirements we are looking for within our adversarial model.

A secret set is a representation of a subset of users of a given universe, satisfying the following properties: any user of the universe can check whether he is or is not a member of the subset; no one can check if another user of the universe is or is not a member; no one can determine the size of the subset. We formalize this primitive as follows:

Definition 3.1 *Let* $\mathcal{U} = \{u_1, \ldots, u_n\}$ *be a universe of users. A* secret set scheme Σ=$(Kgen, Srep, Mver)$ *for* $\mathcal{U}$ *in the public key setting is a triple of algorithms satisfying the following conditions:*

1. *The key-generation algorithm* $Kgen$ *takes as input the security parameter* 1^k, *and outputs a sequence of pairs* $(pub_1, sec_1), \ldots, (pub_n, sec_n)$ *for users* $u_1, \ldots, u_n$.

2. *The set-representation algorithm* $Srep$ *takes as input the description of a set* $S = \{u_{i_1}, \ldots, u_{i_s}\} \subseteq \mathcal{U}$ *and the public information* $pub_S = (pub_{i_1}, \ldots, pub_{i_s})$, *and outputs a string* $SR \in \{0,1\}^{poly(k)}$, *the secret set representation.*

3. *The membership-verification algorithm $Mver$ takes as input SR, a secret set representation generated by $Srep()$ on input a set S and pub_S, and the public and secret information pub_i and sec_i. It outputs 1 if $u_i \in SR$, 0 if $u_i \notin SR$, and $\perp$ in case of failure.*

For $i = 1, \ldots, n$, denoting with $m_i \in \{0,1\}$ the membership status of user u_i in S, i.e., $m_i = 0$ if $u_i \notin S$, and 1 otherwise, it is required that, for each $S \subseteq \mathcal{U}$, there exists a negligible function $negl(k)$ such that:

$$Pr[((pub_1, sec_1), \ldots, (pub_n, sec_n)) \leftarrow Kgen(1^k);\ SR \leftarrow Srep(S, pub_S) : \\ Mver(SR, pub_i, sec_i) = m_i] \geq 1 - negl(k).$$

Basically, the definition states that any user can correctly check its membership status with overwhelming probability of getting the right answer. The $Mrev$ algorithm might fail if SR is not a secret set representation. Moreover, notice that, *any one* can construct, by using the publicly available information $pub_1, \ldots, pub_n$, a secret set.

The security requirements of a secret set scheme can be modeled through the following experiments, run by a challenger C and played by adversary $\mathcal{A}$. The first one, referred to as experiment $CFREE_{\Sigma,\mathcal{A}}$ (collusion-free), models collusion attacks in order to find out the membership status of another user.

Experiment 3.2 $CFREE_{\Sigma,\mathcal{A}}$

1. *C runs $Kgen(1^k)$ and publishes $pub_1, \ldots, pub_n$.*
2. *For $poly(k)$ times, $\mathcal{A}$ hands secret set representations SR for arbitrary $S \subseteq \mathcal{U}$ to C, and asks C, for users $u_j \in \mathcal{U}$,* membership verification queries *and* secret information queries*. As replies to the queries of the first type, $\mathcal{A}$ gets the membership status m_j for u_j w.r.t. SR, to the second the secret information sec_j of u_j.*
3. *$\mathcal{A}$ chooses a user $u_i \in \mathcal{U}$, for which no secret information query to C was asked before, and indicates it to C.*
4. *C chooses $b \leftarrow \{0,1\}$ and a set $S \subseteq \mathcal{U}$, where u_i membership in S is defined as $m_i = b$, and constructs a secret set representation SR^* for S.*
5. *For $poly(k)$ times, $\mathcal{A}$ keeps passing secret sets representations SR to C, asking membership verification queries for user $u_j \in \mathcal{U}$, with the restriction $u_j \neq u_i$ if $SR = SR^*$, and secret information queries for users $u_j \neq u_i$. Queries are answered by C as in step 2.*
6. *Finally, $\mathcal{A}$ outputs a guess bit b'.*
7. *If $b' = b$, then C outputs 1; otherwise 0.*

Definition 3.3 *A secret set scheme Σ=($Kgen$, $Srep$, $Mver$) for $\mathcal{U}$ in the public key setting is* membership private *if, for all probabilistic polynomial time adversaries $\mathcal{A}$, there exists a negligible function $negl(k)$ such that:*

$$Pr[CFREE_{\Sigma,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

The second experiment, referred to as the $SHIDE_{\Sigma,\mathcal{A}}$ (size-hiding) experiment, models the size-hiding property of the secret set.

Experiment 3.4 $SHIDE_{\Sigma,\mathcal{A}}$

1. *C runs* $Kgen(1^k)$ *and publishes* $pub_1, \ldots, pub_n$.
2. *For* $poly(k)$ *times,* $\mathcal{A}$ *hands secret set representations* SR *for arbitrary* $S \subseteq \mathcal{U}$ *to C, and asks C, for users* $u_j \in \mathcal{U}$, membership verification queries *and* secret information queries.
3. $\mathcal{A}$ *chooses two sets* S_0 *and* S_1 *such that* $|S_0| \neq |S_1|$ *and no secret information queries have been asked before for users in the symmetric difference subset* $S_0 \Delta S_1 = (S_0 \setminus S_1 \cup S_1 \setminus S_0)$.
4. C *chooses* $b \leftarrow \{0,1\}$ *and constructs a secret set representation* SR^* *for* S_b. *Then,* C *hands* SR^* *to* $\mathcal{A}$.
5. *For* $poly(k)$ *times,* $\mathcal{A}$ *keeps passing secret sets representations* SR *to C, asking membership verification queries for user* $u_j \in \mathcal{U}$, *with the restriction* $u_j \notin S_0 \Delta S_1$ *if* $SR = SR^*$, *and secret information queries for users* $u_j \notin S_0 \Delta S_1$.
6. *Finally,* $\mathcal{A}$ *outputs a guess bit* b'.
7. *If* $b' = b$, *then C outputs* 1*; otherwise* 0.

Definition 3.5 *A secret set scheme* Σ=$(Kgen,\ Srep,\ Mver)$ *for* $\mathcal{U}$ *in the public key setting is* size hiding *if, for all probabilistic polynomial time adversaries* $\mathcal{A}$, *there exists a negligible function* $negl(k)$ *such that:*

$$Pr[SHIDE_{\Sigma,\mathcal{A}} = 1] \leq \frac{1}{2} + negl(k).$$

Remark. The adversary we have considered in defining experiments $CFREE_{\Sigma,\mathcal{A}}$ and $SHIDE_{\Sigma,\mathcal{A}}$ is an *adaptive* adversary, which can mount *chosen ciphertext* attacks. It is a powerful adversary. We will also consider a weaker adversary, a *non-adaptive* adversary, which is *not allowed* to ask *post-challenge* queries, and we will refer to the corresponding notions as to weak notions. Notice that the class of such adversaries *includes* the class of *static* adversaries, which choose *at the beginning* of the experiments which users to corrupt and get the corresponding secret information.

4 Constructions

In this section we describe some secret set constructions, and prove the properties they enjoy according to the definitions provided in the previous section. Our proof techniques are similar to those recently employed in [13].

Let us start with a light revisitation of the first construction proposed in [15]. Roughly speaking, SR is an authenticated sequence of encryptions, one for each $u \in \mathcal{U}$, of a string which specifies whether $u_i \in S$ or $u_i \notin S$. More precisely:

PK-based Construction.

Let $\Pi = (eGen, Enc, Dec)$ be a public key encryption scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, and let $in, out \in \mathcal{M}$ be public strings. Moreover, let $\Sigma = (sGen, Sign, Ver)$ be digital signature scheme, and let $S = \{u_{i_1}, \ldots, u_{i_s}\} \subseteq \{u_1, \ldots u_n\} = \mathcal{U}$.

- $KGen(1^k)$:
  1. For $i = 1, \ldots, n$, generate $(pk_i, sk_i) \leftarrow eGen(1^k)$.
  2. Set $pub_i = pk_i$ and $sec_i = sk_i$.
- $Srep(S, pub_S)$:
  1. Generate $(vk, sk) \leftarrow sGen(1^k)$.
  2. For $j = 1, \ldots, n$, if $u_j \in S$, then set $c_j = Enc_{pk_j}(in||vk)$, else set $c_j \leftarrow Enc_{pk_j}(out||vk)$.
  3. Compute $\sigma \leftarrow Sign_{sk}(c_1|| \ldots ||c_n)$.
  4. Output $SR = [(c_1, \ldots, c_n), \sigma]$.
- $Mver(SR, pub_i, sec_i)$: Parse SR as a sequence of n values in $\mathcal{C}$ plus a signature. If fail, then output $\perp$. Otherwise,
  1. Set $m = Dec_{sk_i}(c_i)$.
  2. If $m \stackrel{?}{=} \perp$ then output $\perp$; otherwise, if $m \stackrel{?}{=} in||vk$ and $Ver_{vk}(c_1|| \ldots ||c_n, \sigma) = 1$ then output 1; else, if $m \stackrel{?}{=} out||vk$ and $Ver_{vk}(c_1|| \ldots ||c_n, \sigma) = 1$ output 0, else output $\perp$.

Theorem 4.1 *The PK-based constrution, assuming $\Pi = (eGen,\ Enc, Dec)$ is a* cca-secure *public-key encryption scheme and $\Sigma = (sGen,\ Sign,\ Ver)$ is an existentially unforgeable under chosen-message attack digital signature scheme, is a* membership private *and* size-hiding *secret set scheme.*

PROOF. We prove that the scheme in membership private by showing that two experiments, G_0 and G_1, are indistinguishable. Experiment G_0 is the experiment in which the challenger C chooses $b = 0$, i.e., $u_i \notin S$. Experiment $G_1$is the experiment in which the challenger C chooses $b = 1$, i.e., $u_i \in S$. The proof is by contradiction. If G_0 and G_1 were distinguishable by an adversary $\mathcal{A}$, then there exists a distinguisher $\mathcal{D}$ which breaks the cca-security of the encryption scheme or a forger $\mathcal{F}$ which breaks the existential unforgeability under chosen message attacks of the digital signature scheme. Technically, we need two more experiments, G_j^{UF}, for $j = 0, 1$, defined as follows:

- G_j^{UF} is as G_j but, denoting by $SR^* = [(c_1^*, \ldots, c_n^*), \sigma^*]$ the challenge secret set representation, each time the challenger C receives a (post-challenge) membership verification query for $SR = [(c_1, \ldots, c_n), \sigma]$ such that $c_t = c_t^*$ for some $t \in \{1, \ldots, n\}$ and $\sigma \neq \sigma^*$, then the challenger replies, *without decrypting*, with $\perp$.

It is easy to check that experiments G_j and G_j^{UF} are distinguishable *only if* there exists a forger $\mathcal{F}$, which is able to produce post challenge secret set representations accepted in G_j and rejected in G_j^{UF}. Moreover, we show that G_0^{UF} and G_1^{UF} are indistinguishable. Indeed, if they were distinguishable

by an adversary $\mathcal{A}$, we can set up a distinguisher for the experiment $Pub^{CCA}_{\Pi,\mathcal{A}}$. The distinguisher $\mathcal{D}$ simulates the challenger C of the $CFREE_{\Sigma,\mathcal{A}}$ experiment and uses $\mathcal{A}$ as a subroutine to win the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment.

Distinguisher $\mathcal{D}$ for the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment.

1. Receive from the challenger of the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment a public key $\mathbf{pk}$.
2. Run $Gen(1^k)$ for $n-1$ times, and publishes the public keys $pk_1, \ldots, pk_{i-1}, \mathbf{pk}, pk_{i+1}, \ldots, pk_n$.
3. For $poly(k)$ times, reply to $\mathcal{A}$'s membership verification queries and secret information queries of users $u_j \in \mathcal{U}$, for secret set representations SR of arbitrary $S \subseteq \mathcal{U}$. Queries for users for which $\mathcal{D}$ has the secret keys are handled directly, following the scheme. Membership verification query for u_i are answered by sending the ciphertext to the oracle available in the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment, and returning 0 or 1, depending on the oracle's decryption, following the scheme. A secret information query for u_i implies abort with failure.
4. If $\mathcal{A}$ chooses user $u_j \neq u_i$, then abort with failure.
5. Compute $(vk, sk) \leftarrow sGen(1^k)$, send to the $Pub^{CCA}_{\Pi,\mathcal{A}}$ challenger the messages $out||vk$ and $in||vk$, and store the ciphertext c^*, obtained as a reply from the $Pub^{CCA}_{\Pi,\mathcal{A}}$ challenger.
6. Construct, according to the scheme, a secret set representation SR^* for a set $S \subseteq \mathcal{U}$, where u_i membership in S is represented by c^*.
7. For $poly(k)$ times, reply to $\mathcal{A}$'s membership verification queries for secret set representations SR of arbitrary $S \subseteq \mathcal{U}$ for users $u_j \in \mathcal{U}$, and to secret information queries for users $u_j \neq u_i$. The queries are asked applying the same strategy applied before.
8. Finally, forward the bit b' that $\mathcal{A}$ outputs.

Notice that, if the $Pub^{CCA}_{\Pi,\mathcal{A}}$ challenger encrypts $out||vk$, then the view $\mathcal{A}$ gets is identically distributed to the view obtained by running in experiment G_0^{UF}. Otherwise it is identically distributed to the view obtained by running in G_1^{UF}. It is easy to check that if $\mathcal{A}$ distinguishes G_0^{UF} from G_1^{UF} (and, hence, breaks the membership-private property of the scheme) with non negligible probability $\epsilon(k)$, then $\mathcal{D}$ breaks the cca-security property of the scheme with probability $\epsilon(k)/poly(k)$, which is still non negligible.

The size-hiding property can be shown by using a similar strategy. We define experiment G_0 as the experiment in which the challenger chooses the bit $b = 0$, and G_1 as the experiment in which the challenger chooses the bit $b = 1$. Also G_0^{UF} and G_1^{UF} are defined as before. Moreover, we define ℓ hybrid experiments $\mathcal{H}_i$, were $\mathcal{H}_0 = G_0^{UF}$, $\mathcal{H}_\ell = G_1^{UF}$, and $\mathcal{H}_i$ differs from $\mathcal{H}_{i-1}$ *exactly in one ciphertext* in the secret set representation SR^*, which encrypts the message $in||vk$ in $\mathcal{H}_{i-1}$ and $out||vk$ in $\mathcal{H}_i$ or vice versa. More precisely, if $S_0 = \{v_1, \ldots, v_s\}$ and $S_1 = \{w_1, \ldots, w_t\}$, and w.l.o.g., for $\rho < s$, it holds that $v_1 = w_1, \ldots, v_\rho = w_\rho$, then the sets $H_0, \ldots, H_\ell$, used in the hybrid experiments are defined as $H_0 = \{v_1, \ldots, v_s\}, H_1 = \{v_1, \ldots, v_{s-1}\}, \ldots, H_{|S_0 \cap S_1|} = \{v_1, \ldots, v_\rho\}, \ldots, H_{\ell-1} = \{v_1, \ldots, v_s, w_{s+1}, \ldots, w_{t-1}\}, H_\ell = \{w_1, \ldots, w_t\}$.

Arguing by contradiction, it follows that, if G_0^{UF} and G_1^{UF} were distinguishable by an adversary $\mathcal{A}$ with non negligible probability $\epsilon(k)$, then there exists a distinguisher $\mathcal{D}$ which distinguishes between $\mathcal{H}_i$ and $\mathcal{H}_{i-1}$ for a certain i with non negligible probability at least $\epsilon(k)/\ell$. But $\mathcal{D}$ can be used as a subroutine to set up a distinguisher $\mathcal{B}$, which breaks the cca-security property of the encryption scheme with non negligible probability at least $\epsilon(k)/poly(k)$. □

Notice that the PK-based construction is *inefficient* in terms of representation length of the secret set. We can trade-off the representation length as follows: we remove all $Enc_{pk_j}(out||vk)$. The resulting construction *is not* size-hiding, but it has representation length linear in $|S|$ instead of $|\mathcal{U}|$. Each user, to verify its membership status in the secret set, needs to decrypt $|S|$ elements until the right one, if any, is found and decrypted. Unfortunately, we are unable to prove that such a construction is membership private. Indeed, such a property seems impossible to achieve with the above design strategy. An adaptive adversary $\mathcal{A}$ playing experiment $CFREE_{\Sigma,\mathcal{A}}$, can *always* ask post challenge queries for *all* users $u_j \neq u_i$ and, by using the size of S which the construction leaks, it can figure out whether $u_i \in S$ or not. Solutions like adding encryptions for *dummy users* or fixing an a-priori upper bound to the size of the set, do not work. An adversary $\mathcal{A}$ has always a non-negligible probability of guessing whether $u_i \in S$. Nevertheless, we prove that the construction, which can be further simplified gaining in efficiency, is weak membership private, i.e., membership private against a non-adaptive adversary[1].

Representation-length-efficient PK-based constrution.

Let $\Pi = (eGen, Enc, Dec)$ be a public key encryption scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, and let $in \in \mathcal{M}$ be a public string. Moreover, let $S = \{u_{i_1}, \ldots, u_{i_s}\} \subseteq \{u_1, \ldots u_n\} = \mathcal{U}$.

- $KGen(1^k)$:
  1. For $i = 1, \ldots, n$, generate $(pk_i, sk_i) \leftarrow eGen(1^k)$.
  2. Set $pub_i = pk_i$ and $sec_i = sk_i$.
- $Srep(S, pub_S)$:
  1. For $j = 1, \ldots, s$, set $c_j = Enc_{pk_{i_j}}(in||u_{i_j})$.
  2. Output $SR = (c_1, \ldots, c_s)$.
- Parse SR as a sequence of s values in $\mathcal{C}$. If fail, then output $\perp$. Otherwise, for $j = 1, \ldots, s$
  1. Set $m = Dec_{sk_i}(c_j)$.
  2. If $m \stackrel{?}{=} in||u_i$ then output 1; else, if $j \stackrel{?}{=} s$ then output 0.

However, in order to prove that the construction is weak membership-private, we *need to require* the public key encryption scheme to satisfy the *key-privacy* and *weak robustness* properties. Roughly speaking, the first one guarantees that each piece of the secret set representation does not leak any information about the public key with whom it has been computed, while the second guarantees that a ciphertext can be correctly decrypted only with a single decryption key.

Theorem 4.2 *The representation-length-efficient PK-based construction, assuming $\Pi = (Gen, Enc, Dec)$ is a ik-cca* private *and* weakly robust *public-key encryption scheme, is a* weak membership private *secret set.*

[1]Notice that we could also prove a stronger result: the construction, amended with digital signatures, is membership private against an adaptive adversary which *does not* ask membership queries on the challenge representation SR^*. However, since such a notion is a non-standard notion, we limit our discussion to non-adaptive adversaries, as usually defined in the literature, (e.g., see the CCA1 security notion for public-key encryption schemes in [3])

PROOF. Sketch. We prove the theorem by showing that the following three experiments, run on sets of the same size, are indistinguishable.

- G_0 is the experiment in which the challenger C chooses the bit $b = 0$, i.e., $u_i \notin S$. Let us denote S with S_0 and with v the user which replaces u_i. S_0 has size s.
- G_0^{WR-KP} is the experiment in which the challenger C replaces the encryption for user v with an encryption for v with the public key of u_i.
- G_1 is the experiment in which the challenger C chooses the bit $b = 1$, i.e., $u_i \in S$. Let us denote S with S_1. The set S_1 has size s. User u_i replaces user v in S_0.

By using similar arguments to the ones employed in the previous proof, we show that G_0 and G_0^{WR-KP} are indistinguishable assuming the public-key encryption scheme is key-private. Precisely, a distinguisher for the two experiments could be used to win the ik-cca experiment. At the same time, G_0^{WR-KP} and G_1 are indistinguishable assuming the public-key encryption scheme is weakly robust. Indeed, if they were not, $\mathcal{A}$, by using the secret key of user v could check whether user u_i is in S or not and succeed with non-negligible probability. □

The following construction, introduced in [15], exhibits a trade-off between security and efficiency.

DH-based Bit-Vector Construction.

Let G be a cyclic group of order q with generator g. Moreover, let $S = \{u_{i_1}, \ldots, u_{i_s}\} \subseteq \{u_1, \ldots u_n\} = \mathcal{U}$.

- $KGen(1^k)$:
  1. For $i = 1, \ldots, n$, set $a_i \leftarrow Z_q^*$ and compute g^{a_i}.
  2. Set $pub_i = g^{a_i}$ and $sec_i = a_i$.
- $Srep(S, pub_S)$:
  1. Choose $b \leftarrow Z_q^*$ and compute g^b.
  2. For $j = 1, \ldots, n$, compute $K_j = (g^{a_i})^b$ and set $c_j = MSB(K_j)$ if $u_j \in S$ or $c_j = MSB(K_j) + 1 \bmod 2$ if $u_j \notin S$.
  3. Output $SR = (g^b, c_1 \ldots c_n)$.
- $Mver(SR, pub_i, sec_i)$:
  1. Compute $K_i = (g^b)^{a_i}$ and $d_i = MSB(K_i)$.
  2. If $d_i \stackrel{?}{=} c_i$, then return 1; otherwise, return 0.

We show that the construction enjoys the *weak* membership private and size hiding properties, i.e., the adversary is non adaptive. We need the *Computational Diffie Hellman* assumption and the concept of *hard-core* predicate.

Definition 4.3 *Let (G, q, g) be a cyclic group of order q with generator g. Let $a, b \leftarrow Z_q$, and define $DH_g(g^a, g^b) = g^{ab}$. The Computational Diffie Hellman problem (for short,* CDH problem*) consists in computing $DH_g(g^a, g^b)$.*

The hardness of the CDH problem in G is formalised as follows:

Definition 4.4 *The* CDH problem *is hard relative to G if, for all probabilistic polynomial time algorithms $\mathcal{A}$, there exists a negligible function $negl(k)$ such that*

$$Pr[a, b \leftarrow Z_q : \mathcal{A}(G, g, q, g^a, g^b) = g^{ab}] \leq negl(k)$$

Hard-core predicates are defined as follows:

Definition 4.5 *A function $hc : \{0,1\}^* \rightarrow \{0,1\}$ is a* hard-core predicate *of a function f if hc can be computed in polynomial time but, for every probabilistic polynomial-time algorithm $\mathcal{A}$, there exists a negligible function $negl(k)$ such that*

$$Pr[x \leftarrow \{0,1\}^k : \mathcal{A}(f(x)) = hc(x)] \leq 1/2 + negl(k).$$

It is well known (e.g., see [11]) that the $MSB(g^{ab})$ is a hard-core predicate for the function

$$f : (a, b) \in Z_q \times Z_q \rightarrow (g^a, g^b) \in G \times G.$$

Theorem 4.6 *The DH-based Bit-Vector construction, assuming the CDH problem is hard in G and $MSB(g^{ab})$ is a hard-core predicate for f, is a weak* membership private *and* size hiding *secret set.*

PROOF. Sketch. We define experiments G_0 and G_1 as before. Arguing by contradiction, if G_0 and G_1 were distinguishable by a distinguisher $\mathcal{D}$, it is because of two possible reasons: either $\mathcal{D}$ distinguishes because it computes $g^{a_i b}$ from the publicly available g^{a_i} and g^b and, hence, gets $MSB(g^{a_i b})$. But such an event contradicts the assumption that the computational Diffie Hellman problem is difficult in G. Or, $\mathcal{D}$ distinguishes with non negligible probability the elements in the i-th position in the two n-bit strings representing the set S in the two experiments. However, in such a case, $\mathcal{D}$ can be used as a subroutine, by a properly designed algorithm $\mathcal{A}$, to guess $MSB(g^{a_i b})$ with non negligible probability, as follows:

Predictor $\mathcal{A}$ for the hard-core predicate $MSB(g^{ab})$.

1. Receive in input g^a and g^b.
2. Choose an index $i \in \{1, \ldots, n\}$.
3. Choose $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \leftarrow Z_q$, compute $g^{a_1}, \ldots, g^{a_{i-1}}, g^{a_{i+1}}, \ldots, g^{a_n}$ Publish the public keys $g^{a_1}, \ldots, g^{a_{i-1}}, g^a, g^{a_{i+1}}, \ldots, g^{a_n}$.
4. For $poly(k)$ times, reply to $\mathcal{D}$'s membership verification queries and secret information queries of users $u_j \in \mathcal{U}$, for secret set representations SR of arbitrary $S \subseteq \mathcal{U}$. Queries for users for which $\mathcal{D}$ has the secret keys are handled directly. A membership verification query or a secret information query for u_i implies abort with failure.
5. If $\mathcal{D}$ chooses user $u_j \neq u_i$, then abort with failure.
6. Choose $b^* \leftarrow \{0,1\}$.

7. Construct, according to the scheme and using g^b, a secret set representation SR^* for a random set $S \subseteq \mathcal{U}$, where u_i membership in S is represented by b^*, and hand it to $\mathcal{D}$.
8. Finally, if $\mathcal{D}$ outputs 1, then output b^*; otherwise, output $1 - b^*$.

Notice that if $b^* = MSB(g^{a_i b})$, than $\mathcal{D}$ is playing experiment G_1; otherwise, $\mathcal{D}$ is playing experiment G_0. Therefore, if $\mathcal{D}$ distinguishes G_0 from G_1 with non negligible probability $\epsilon(k)$, then the predictor algorithm $\mathcal{A}$ computes $MSB(g^{a_i b})$ with probability at least $\epsilon(k)/n$.

Similarly, the size hiding property can be shown by using the same strategy but with a sequence of hybrid experiments $\mathcal{H}_0, \ldots, \mathcal{H}_\ell$, where $\mathcal{H}_0 = G_0$, $\mathcal{H}_\ell = G_1$, and $\mathcal{H}_i$ and $\mathcal{H}_{i-1}$ differs in exactly one position of the n-bit string. Arguing by contradiction, if G_0 and G_1 were distinguishable with non-negligible probability by a distinguisher $\mathcal{D}_1$, then there exists a distinguisher $\mathcal{D}_2$ which distinguishes, for a certain i, with a polynomially related non-negligible probability between $\mathcal{H}_i$ and $\mathcal{H}_{i-1}$. The distinguisher $\mathcal{D}_2$ can be used as in the above sketch for proving membership privateness, in order to break the hard-coreness of $MSB(g^{ab})$. □

Remark. Notice that we do not know how to (amend and) prove the security against an adaptive adversary because, with our proof technique, we *cannot simulate* post-challenge membership verification queries for user u_i (i.e., we do not know the secret exponent a and we have no supporting oracle, like in the $Pub^{CCA}_{\Pi,\mathcal{A}}$ experiment). Hence, $\mathcal{A}$ can easily distinguishes an execution within the $CFREE_{\Sigma,\mathcal{A}}$ experiment from one simulated by $\mathcal{D}$.

Let us denote by $H : \{0,1\}^* \rightarrow \{0,1\}^k$ an *hash function*. An interesting variant of the above construction, efficient and secure in the random oracle model [12], is the following:

Hash-based Construction.

Let G be a cyclic group of prime order q with generator g. Moreover, let $S = \{u_{i_1}, \ldots, u_{i_s}\} \subseteq \{u_1, \ldots u_n\} = \mathcal{U}$.

- $KGen(1^k)$:
  1. For $i = 1, \ldots, n$, set $a_i \leftarrow Z_q^*$ and compute g^{a_i}.
  2. Sets $pub_i = g^{a_i}$ and $sec_i = a_i$.
- $Srep(S, pub_S)$:
  1. Choose $b \leftarrow Z_q^*$ and compute g^b.
  2. For $j = 1, \ldots, s$, compute $K_{i_j} = (g^{a_{i_j}})^b$ and set $c_j = H(K_{i_j})$.
  3. Output $SR = [g^b, c_1, \ldots, c_s]$.
- $Mver(SR, pub_i, sec_i)$:
  1. Compute $K_i = (g^b)^{a_i}$ and $h = H(K_i)$.
  2. If $h \in \{c_1, \ldots, c_s\}$, then return 1; otherwise, return 0.

By looking at the private broadcast encryption schemes proposed in [4], the reader may notice that actually the above scheme is 'contained' in the scheme proposed in subsection 4.2 of [4]. Indeed, in

order to enable efficient decryption, the authors of [4], add to the broadcast message a sequence of hints: it is not difficult to see that these hints represent a hidden representation of the set of recipients, which, in our language is a secret set.

Theorem 4.7 *The Hash-based construction, assuming the computational Diffie Hellman problem is difficult in G, is a* weak membership private *secret set in the random oracle model.*

PROOF. Sketch. Arguing by contradiction, if the experiments G_0 and G_1 were distinguishable by a distinguisher $\mathcal{D}$, it is because $\mathcal{D}$ computes $g^{a_i b}$ from the publicly available g^{a_i} and g^b. Indeed, since H is a random oracle, then $H(x)$ is a truly random value, and the only possibility $\mathcal{D}$ has to distinguish G_0 from G_1 is to compute x. But such an event contradicts the assumption that the CDH problem is difficult in G. From a technical point of view, if $\mathcal{D}$ exists, then we can construct an adversary $\mathcal{A}$ which, step by step, acts like $\mathcal{D}$ until it computes g^{ab}. □

5 Key-privacy and security under chosen ciphertext attacks

In our representation-length efficient construction, and in other applications (e.g., [16]), the message which is encrypted *is known* and *does not* need to be protected. Thus, of general interest is the question whether there exists an ik-cca public-key encryption scheme which is not cca-secure. Indeed, in [2], key-privacy was studied as an *additional* property for cca-secure public-key encryption schemes but the above issue was not investigated. Here we show that, in order to be ik-cca secure, a public key encryption scheme has to be *non-malleable* [8]. Loosely speaking, a public key encryption scheme is non-malleable if, given an encryption $c = E_{pk}(m)$, it is not feasible to produce a new ciphertext c' which is an encryption of a message m', somehow related to m. Formally, non-malleability is defined through the following experiment[2], parameterized by $b \in \{0, 1\}$, run by a challenger C:

Experiment 5.1 $Exp^{nm-b}_{\mathcal{A},\Pi}$

1. $(pk, sk) \leftarrow Gen(1^k)$.
2. $(M, s) \leftarrow \mathcal{A}^{O_{sk}}(pk)$.
3. $x_0, x_1 \leftarrow M; \quad y \leftarrow Enc_{pk}(x_1)$;
4. $(R, \mathbf{z}) \leftarrow \mathcal{A}^{O_{sk}}(M, s, y)$;
5. $\mathbf{x} \leftarrow Dec_{sk}(\mathbf{z})$;
6. *If* $y \notin \mathbf{z} \wedge \perp \notin \mathbf{x} \wedge R(x_b, \mathbf{x})$, *then C outputs* 1; 0 *otherwise.*

In bold we have denoted vectors of elements, and Dec_{sk} on vectors is computed by applying the Dec_{sk} operation componentwise. Experiments $Exp^{nm-b}_{\mathcal{A},\Pi}$ works as follows. The challenger C generates a key-pair (pk, sk) and hands the public key pk to the adversary. The adversary, with access to the decryption oracle, outputs an algorithm M for sampling elements from the domain $\mathcal{M}$ of the encryption scheme and some state information s. The challenger C uses M to draw two elements x_0, x_1 of the same lenght, and computes an encryption y of x_1. Then the adversary, using M, s and y, and with access to the decryption oracle (which cannot be invoked on y), outputs a description of a relation R,

[2]The definition we use comes from [3] since it was shown to be equivalent (but easier to work with) to the simulation-based definition given in [8].

computable in polynomial time, and a vector of ciphertexts $\mathbf{z}$. Finally, the challenger C decrypts one by one the ciphertexts in $\mathbf{z}$, obtaining a vector of plaintexts $\mathbf{x}$ and, if $y \notin \mathbf{z}, \perp \notin \mathbf{x}$ and $R(x_b, \mathbf{x})$ holds, then it outputs 1. Notice that, if the bit b which parameterizes $Exp^{nm-b}_{\mathcal{A},\Pi}$ is equal to 1 and C outputs 1, the adversary has produced encryptions $\mathbf{z}$ based on y which contain plaintexts in relation with x_1. On the other hand, when the bit b is equal to 0 and C outputs 1, the adversary has produced encryptions $\mathbf{z}$ whose plaintexts $\mathbf{x}$ are in relation with x_0, which is independent of y. Let

$$Adv^{nm}_{\mathcal{A},\Pi} = Prob[Exp^{nm-1}_{\mathcal{A},\Pi} = 1] - Prob[Exp^{nm-0}_{\mathcal{A},\Pi}].$$

The advantage $Adv^{nm}_{\mathcal{A},\Pi}$ quantifies *the gap* between the cases in which $\mathcal{A}$ has success in the experiment by producing new encryptions based on y whose plaintexts are related to x_1 and the cases in which it just happens by accident that $\mathcal{A}$ produces encryptions which are related to some randomly chosen x_0.

Definition 5.2 *Let $\Pi = (Gen, Enc, Dec)$ be a public-key encryption scheme. Π is non-malleable if, for all probabilistic polynomial time algorithms $\mathcal{A}$, there exists a negligible function $negl(k)$ such that*

$$Adv^{nm}_{\mathcal{A},\Pi} \leq negl(k);$$

By using the above definition, we prove that an encryption scheme, to be ik-cca secure, needs to be non-malleable. The idea of the proof is that, if a scheme is malleable, an adversary against key-privacy can always modify the challenge ciphertext c^*, obtained in the $Exp^{ik\text{-}cca}_{\mathcal{A},\Pi}$ experiment on input m, to get a new ciphertext c' which can be provided as input to the decryption oracle, and then check whether the plaintext m' received from the oracle is in relation with m, according to one of the two public keys pk_0 and pk_1. More precisely, we prove the following result:

Theorem 5.3 *Let $\Pi = (Gen, Enc, Dec)$ be a public-key encryption scheme. Π is ik-cca secure only if Π is non-malleable.*

PROOF. Sketch. Let us assume that Π is malleable and let $\mathcal{A}^{nm}$ be an adversary who, given an encryption c^* of message m, yields encryptions $\mathbf{c}$ of related messages $\mathbf{x}$. By using $\mathcal{A}^{nm}$ as a subroutine, we set up an adversary $\mathcal{A}^{ik\text{-}cca}$, which works as follows:

1. Receive pk_0 and pk_1.
2. Run two instances of $\mathcal{A}^{nm}$, say $\mathcal{A}^{nm}_0$ and $\mathcal{A}^{nm}_1$, providing pk_0 to the first and pk_1 to the second. and simulate the environment in which the two instances run. Let (M_0, s_0) and (M_1, s_1) be the two outputs for the first phase of $Exp^{nm-0}_{\mathcal{A},\Pi}$ from $\mathcal{A}^{nm}_0$ and of $Exp^{nm-0}_{\mathcal{A},\Pi}$ from $\mathcal{A}^{nm}_1$.
3. Choose an m sampleable with both M_0 and M_1, and hand m to the ik-cca challenger, getting $c^* = Enc_{pk_b}(m)$.
4. Hand c^* to $\mathcal{A}^{nm}_0$ and $\mathcal{A}^{nm}_1$, and receive[3] $(R_0, \mathbf{c}_0)$ and $(R_1, \mathbf{c}_1)$.
5. Ask oracle O^{sk_0} to decrypt componentwise $\mathbf{c}_0$ and oracle O^{sk_1} to decrypt componentwise $\mathbf{c}_1$, obtaining $\mathbf{x}_0$ and $\mathbf{x}_1$.

[3]One of the two adversaries might not reply if it somehow notices the encryption c^* has been produced with an encryption key different from the one received in input. In such a case, $\mathcal{A}^{ik\text{-}cca}$ easily wins.

6. If $R_0(m, \mathbf{x}_0)$ holds, then output 0; else, if $R_1(m, \mathbf{x}_1)$ holds, then output 1.

□

It is easy to check that $\mathcal{A}^{ik\text{-}cca}$'s advantage in winning the key-privacy experiment is exactly the same advantage $\mathcal{A}^{nm}$ has in winning the malleability experiment. Moreover, since the notion of non-malleability is *equivalent* to cca-security [3, 8], we get the following result:

Corollary 5.4 *Let* $\Pi = (Gen, Enc, Dec)$ *be a public-key encryption scheme.* Π *is* ik*-cca secure only if* Π *is* cca*-secure.*

6 Conclusions and Open Problems

We have proposed a *model* for secret sets in the public key setting, and we have presented some *constructions*, which are a revisitation of previously proposed constructions. The novelty of the present work is the security analysis within the model. Basically, we have shown that:

1. A cca-secure public-key encryption scheme and an existentially unforgeable under chosen-message attack digital signature scheme are *sufficient* to design a membership-private size-hiding secret set. The construction is not length efficient.

2. A length-efficient weak membership private but not size-hiding construction can be obtained by using a weakly robust and key-private public-key encryption scheme.

3. Using a group in which the CDH problem is hard and assuming the $MSB(\cdot)$ predicate is hard-core, we get a weak membership-private size-hiding secret set which is sort of length-efficient.

4. Using a group in which the CDH problem is hard can be obtained a length-efficient weak membership-private secret set representation, secure in the random oracle model.

Several problems are still open. The most intriguing one is: does exist a length-efficient membership-private and size-hiding secret set construction? The one we have described has length $O(|\mathcal{U}|)$. Alternatively, does exist a lenght-efficient membership private secret set construction? The one we have described is only weak membership private. Actually, length efficiency seems difficult to achieve, i.e., to come up with a construction which does not involve the whole universe of users and does not leak an upper bound on the set size. Such a property has been obtained in the context of zero-knowledge sets and, as the authors pointed out [14], it was the most challenging property to achieve in their construction.

References

[1] M. Abdalla, M. Bellare, and G. Neven, *Robust Encryption*, Proc. of TCC 2010, LNCS, Vol. 5978, pp. 480-497, 2010.

[2] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval, *Key-Privacy in Public-Key Encryption*, Proc. of Asiacrypt 2001, LNCS, Vol. 2248, pp. 566-582, 2001.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations among notions of security for public-key encryption schemes*, Proc. of Crypto 1998, LNCS, Vol. 1462, pp. 26-45, 1998.

[4] A. Barth, D. Boneh and B. Waters, *Privacy in Encrypted Content Distribution Using Private Broadcast Encryption*, Proc. of Financial Cryptography (FC 2006), LNCS, Vol. 4107, pp. 52-64, 2006.

[5] D. Catalano, Y. Dodis and I. Visconti, *Mercurial Commitments: Minimal Assumptions and Efficient Constructions*, Proc. of the third Theory of Cryptography Conference (TCC 06), LNCS Vol. 3876, pp. 120-144, 2006.

[6] D. Catalano, D. Fiore and M. Messina, *Zero-Knowledge Sets with Short Proofs*, Proc. of Eurocrypt 2008, LNCS, Vol. 4965, pp. 433-450, 2008.

[7] M. Chase, A. Healy, A. Lysyanskaya, T. Malkin and L. Rezin, *Mercurial Committments with applications to zero-knowkedge sets*, Proc. of Eurocrypt 2005, LNCS Vol. 3494, pp. 422-439, 2005

[8] D. Dolev, C. Dwork, and M. Naor, *Non-malleable cryptography*, SIAM Journal of Computing, Vol. 30, N. 2, pp. 391-437, 2000.

[9] A. De Santis and B. Masucci, *On Secret Set Schemes*, Inform. Process. Lett. 74 (2000) 243-251.

[10] G. Goldwasser and S. Micali, *Probabilistic Encryption*, Journal of Computer and System Sciences 28(2), 278-299 (1984).

[11] J. Hastad and M. Naslund, *The Security of all RSA and Discrete Log Bits*, Journal of ACM, Vol. 51, N. 2, 2004.

[12] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, 2008.

[13] B. Libert, K. Paterson, and E. A. Quaglia, *Anonymous Broadcast Encryption*, eprint archive, 2011.

[14] S. Micali, M. Rabin and J. Kilian, *Zero-Knowledge Sets*, Proc. of the 44th IEEE Symposium on Foundations of Computer Science (FOCS 2003).

[15] R. Molva and G. Tsudik, *Secret Sets and Applications*, Inform. Process. Lett. 65 (1998) 47-55.

[16] K. Sako, *An auction protocol which hides bids of losers*, Proc. of the Third International Workshop on Theory and Practice of Public-Key Cryptography (PKC 2000), LNCS Vol. 1751, pp. 422-432, 2000.