Relations Among Notions of Plaintext Awareness

James Birkett and Alexander W. Dent

Information Security Group,
Royal Holloway, University of London,
Egham, TW20 0EX, UK.
{j.m.birkett,a.dent}@rhul.ac.uk

Abstract. We introduce a new notion of plaintext awareness in the standard model (PA2I), and prove relations between various security and plaintext awareness properties. In particular, we show that for schemes which are IND-CPA, our new notion is equivalent to the standard notion. Teranishi and Ogata show [10] that a scheme which is OW-CPA and PA2 is IND-CCA2. We show that the analagous result about PA2I does not hold, and will extend their result to show that any scheme which is OW-CPA and PA2 hides the length of a message, and thus must have a finite message space. We also give a formal proof of the conjecture of Dent [6] that if a scheme is PA2 then it is PA1+. Finally, using Dent's ideas [6] we show that the Kurosawa-Desmedt scheme is PA2.

1 Introduction

Plaintext awareness was first defined in the random oracle model by Bellare and Rogaway in [3] as a device to prove the IND-CCA2 security of the RSA-OAEP scheme. The idea behind plaintext awareness is that if an adversary $\mathcal{A}$ (known as a ciphertext creator) can create a valid ciphertext, it must "know" the underlying plaintext.

In their model, $\mathcal{A}$ takes a public key and returns a ciphertext C. They require that there is a "plaintext extractor" K which takes as input C and the list of random oracle queries made by $\mathcal{A}$, and returns the correct decryption of C when given C and the list of random oracle queries made by $\mathcal{A}$.

It was discovered by Shoup [8] that the definition used was too weak to prove the IND-CCA2 security of a scheme, as it did not take into account the adversary's ability to obtain ciphertexts which it did not generate, and thus had never queried the random oracle on the corresponding message. In particular, an IND-CCA2 adversary always gets access to one ciphertext it did not generate – namely the challenge ciphertext.

A stronger notion (still relying on the random oracle model) was introduced by Bellare, Desai, Pointcheval and Rogaway in [1], which added an encryption oracle. This allowed them to prove that a scheme which is IND-CPA and PA is IND-CCA2.

More recently, Bellare and Palacio introduced a notion of plaintext awareness in the standard model [2]. If the original RO model definition was naievely applied in the standard model, a ciphertext extractor could be used to break the IND-CPA security of the scheme. This is because there are no random oracle queries to give it, so the ciphertext extractor would have to decrypt a ciphertext using no information other than the ciphertext itself. Instead, the random coins of the ciphertext creator are given to the ciphertext extractor, allowing it to see how the ciphertext was generated. It is this definition we consider here.

The Bellare-Palacio definition differs from the BDPR defintion in a few other ways. Firstly, the plaintext extractor $\mathcal{A}^*$ is allowed to depend on $\mathcal{A}$. Secondly, instead of simply outputting a ciphertext to be decrypted, the ciphertext creator $\mathcal{A}$ submit ciphertexts to a decryption oracle. The result returned is either the output of the decryption algorithm $\mathcal{D}$ (the real game), or the plaintext extractor $\mathcal{A}^*$ (the fake game). The plaintext returned by $\mathcal{A}^*$ is not required to be identical to the real decryption, only that the output of $\mathcal{A}$ is computationally indistinguishable in the real and fake games.

Thirdly, the encryption oracle was made more powerful. Instead of simply taking a message and encrypting it under the public key, it makes use of a stateful algorithm $\mathcal{P}$, called a plaintext creator. When $\mathcal{A}$ makes an encryption query on some auxiliary information aux, the encryption oracle runs $\mathcal{P}(aux)$, encrypts the result and returns it. This models the idea that an adversary may have partial information about ciphertexts it obtains from a third party, or even partial control over them. In particular, a plaintext creator which we will call $\mathcal{P}_I$ is of particular interest. It takes two messages and outputs one of them at random. This particular plaintext creator is necessary to model the challenge ciphertext in the IND game in the proof that IND-CPA + PA2 implies IND-CCA2.

1.1 Our Contributions

This paper is divided into two parts. In the first part, we investigate the power of the plaintext creator $\mathcal{P}$. We define a modified version of PA2, which we call PA2I, in which the plaintext creator is fixed as $\mathcal{P}_I$, rather than being an arbitrary polynomial-time algorithm. This definition is sufficient to prove that IND-CPA + PA2I implies IND-CCA2, but lacks the generality of PA2. We will show that if a scheme is IND-CPA and PA2I, then it is also PA2. PA2I is conceptually simpler than the full PA2 model and we feel it is sufficient to replace PA2 in most situations.

We also investigate the role of randomness in plaintext awareness. In [6], Dent defined the notion PA1+, in which the ciphertext creator $\mathcal{A}$ may request new random bits. The plaintext extractor $\mathcal{A}^*$ may see all of the random bits used by $\mathcal{A}$ prior to a decryption query being made, but may not see the result of future randomness queries. This prevents $\mathcal{A}^*$ from predicting the output of $\mathcal{A}$. Dent gave evidence that PA2 implies PA1+, but stopped short of a formal proof. We will give a formal proof of this.

Finally, we consider the role of the message space of the encryption scheme. We will show, using the ideas of Teranishi and Ogata [10] that an IND-CPA secure encryption scheme with an infinite message space cannot be PA2 plaintext aware. We note that this proof does not hold for PA2I, but do not prove that there is a secure encryption scheme with an infinite message space which is PA2I.

In the second part, we will prove that the hash proof system variant of the Kurosawa-Desmedt encryption scheme is PA2 plaintext aware, following the strategy that Dent used in [6] to prove the Cramer-Shoup scheme [4] is plaintext aware. Hash proof systems were introduced by Cramer and Shoup in [5] to generalise the Cramer-Shoup encryption scheme of [4], and Kurosawa and Desmedt showed how their scheme can also generalised to use an arbitrary hash proof system.

In order to generalise Dent's the result to the more general framework of hash proof systems based on subset membership problems, we introduce a new extractor assumption, which we call the subset witness knowledge (SWK) assumption. Briefly, a subset membership problem can be thought of as a set X with a subset L. The problem is hard if it is difficult to distinguish a random element of L from a random element of $X \setminus L$. Informally, the SWK assumption states that the only way an algorithm can compute values $x \in L$ is if it also computes a corresponding witness that $x \in L$. We note that the Diffie-Hellman knowledge assumption is simply a special case of the SWK assumption for the Diffie-Hellman problem.

2 Definitions

2.1 Notation

We will use the following notation in this paper. $x \xleftarrow{R} S$ means x is sampled uniformly at random from the set S. For an algorithm $\mathcal{A}$, we write $x \leftarrow \mathcal{A}^{\mathcal{O}}(y, z)$ to mean that x is assigned the output of running $\mathcal{A}$ on inputs y and z, with access to oracle $\mathcal{O}$. If $\mathcal{A}$ is a probabilistic algorithm, we may write $x \leftarrow \mathcal{A}(y, z; R)$ to mean the output of $\mathcal{A}$ when run on input y using the random coins R. If we do not specify R then we implicitly assume that the coins are selected uniformly at random from $\{0,1\}^{\infty}$.

2.2 Statistical and Computational indistinguishability

We will need the statistical distance between two random variables, which we will define as follows: Let x and y be random variables taking values on a finite set S. We define the statistical distance between x and y as

$$\Delta[x, y] = \frac{1}{2} \sum_{s \in S} |\Pr[x = s] - \Pr[y = s]| .$$

We note the following properties of statistical distance:

Let x and y be random variables taking values on a finite set X, and w be a random variable taking values on a set W which is independent of both x and y. Let $f : X \to W$ be a function on X. Let $S(x)$ be a predicate on x. Then

$$\Delta[(x, w), (y, w)] = \Delta[x, y] \tag{1}$$

$$\Delta[f(x), f(y)] \leq \Delta[x, y] \tag{2}$$

$$|\Pr[S(x)] - \Pr[S(y)]| \leq \Delta[x, y] \tag{3}$$

Suppose S, x and y depend on a size parameter λ. We that x and y are statistically indistinguishable if $\Delta[x, y]$ is negligible as a function of λ.

We say two random variables x and y which depend on a size parameter λ are computationally indistinguishable if for any polynomial time algorithm D, $|\Pr[D(x) = 1] - \Pr[D(y) = 1]|$ is negligible as a function of λ. D is known as a distinguishing algorithm.

2.3 Indistinguishability of ciphertexts

We first describe the IND-ATK (where ATK is either CPA or CCA2) game for an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ and $\mathcal{A}_2$ are probabilistic polynomial-time algorithms:

$(pk, sk) \leftarrow \mathcal{G}(1^k)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}}(pk)$
$b \xleftarrow{R} \{0, 1\}$
$C^* \leftarrow \mathcal{E}(pk, m_b)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(C^*, state)$

In the above, $\mathcal{A}_1$ outputs two messages (m_0, m_1) such that $|m_0| = |m_1|$ and some state information. The challenger chooses a bit b at random and encrypts m_b to give a challenge ciphertext C^*. $\mathcal{A}_2$ takes C^* and the state information as input and outputs a guess for b.

We consider two attack models. In the chosen plaintext attack (CPA) model, $\mathcal{A}$ does not have access to any oracles. In the adaptive chosen ciphertext attack (CCA2) model, $\mathcal{A}$ may query a decryption oracle $\mathcal{D}$, which takes a ciphertext C as input and returns $\mathcal{D}(sk, C)$. The only restriction is that $\mathcal{A}_2$ may not query the decryption oracle on C^*.

We define the advantage of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ATK}} = |\Pr[b' = b] - \frac{1}{2}|.$$

Definition 1 (IND-CPA). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is secure in the sense of indistinguishability under chosen plaintext attack if for any IND-CPA adversary $\mathcal{A}$, the advantage $\mathbf{Adv}_{\mathcal{A}}^{IND\text{-}CPA}$ is negligible as a function of k.*

Definition 2 (IND-CCA2). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is secure in the sense of indistinguishability under chosen ciphertext attack if for any IND-CCA2 adversary $\mathcal{A}$, the advantage $\mathbf{Adv}_{\mathcal{A}}^{IND\text{-}CCA}$ is negligible as a function of k.*

2.4 Plaintext Awareness

Definition 3 (PA1). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA1 plaintext aware if for all polynomial-time ciphertext creators $\mathcal{A}$, there exists a polynomial-time plaintext extractor $\mathcal{A}^*$ such that the output x of $\mathcal{A}$ in the real and fake games described below is computationally indistinguishable.*

In both games, the challenger generates a fresh key pair (pk, sk). It then runs $x \leftarrow \mathcal{A}^{\mathcal{O}}(pk)$. $\mathcal{A}$ has access to a decryption oracle which takes a ciphertext C. In the real game, it returns $m \leftarrow \mathcal{D}(sk, C)$. In the fake game, it computes $(m, state) \leftarrow \mathcal{A}^*(C, pk, R[\mathcal{A}], state)$ and returns m.

For a given distinguishing algorithm D, We define the advantage

$$\mathbf{Adv}^{PA1}_{\mathcal{A},\mathcal{A}^*,D} = |\Pr[D(x_{Real}) = 1] - \Pr[D(x_{Fake}) = 1]|$$

where x_{Real} is the output of $\mathcal{A}$ in the Real game and x_{Fake} is the output of $\mathcal{A}$ in the Fake game.

Definition 4 (PA2). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2 plaintext aware if for all polynomial-time ciphertext creators $\mathcal{A}$, there exists a polynomial-time plaintext extractor $\mathcal{A}^*$ such that for all polynomial-time plaintext creators $\mathcal{P}$, the output x of $\mathcal{A}$ in the real and fake games described below is computationally indistinguishable.*

In both games, the challenger generates a fresh key pair (pk, sk) and string $R[\mathcal{A}]$ of random coins for $\mathcal{A}$. It then runs $x \leftarrow \mathcal{A}(pk; R[\mathcal{A}])$. $\mathcal{A}$ has access to an encryption oracle and a decryption oracle. The encryption oracle takes input s, and runs as follows:

$(m, state_{\mathcal{P}}) \leftarrow \mathcal{P}(s, state_{\mathcal{P}})$
$C \leftarrow \mathcal{E}(pk, m)$
Append C to CLIST
Returns C

The decryption oracle takes a ciphertext C not in CLIST*. In the real game, it returns $m \leftarrow \mathcal{D}(sk, C)$, while in the fake game, it computes $(m, state_{\mathcal{A}^*}) \leftarrow \mathcal{A}^*(pk, C, R[\mathcal{A}], \text{CLIST}, state_{\mathcal{A}^*})$ and returns m.*

For a given distinguishing algorithm D, We define the advantage

$$\mathbf{Adv}^{PA2}_{\mathcal{A},\mathcal{A}^*,\mathcal{P},D} = |\Pr[D(x_{Real}) = 1] - \Pr[D(x_{Fake}) = 1]|$$

where x_{Real} is the output of $\mathcal{A}$ in the Real game and x_{Fake} is the output of $\mathcal{A}$ in the Fake game.

Definition 5 (PA2I). *A public key encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2 plaintext aware if for all polynomial-time ciphertext creators $\mathcal{A}$, there exists a polynomial-time plaintext extractor $\mathcal{A}^*$, the output x of $\mathcal{A}$ in the real and fake games described below is computationally indistinguishable.*

In both, the challenger generates a fresh key pair (pk, sk) and random string $R[\mathcal{A}]$. It then runs $x \leftarrow \mathcal{A}(pk; R[\mathcal{A}])$. $\mathcal{A}$ has access to an encryption oracle and a decryption oracle.

The encryption oracle takes input $s = (m_0, m_1)$. The challenger responds to decryption queries as in the case of PA2, and responds to encryption queries as follows:

$b \xleftarrow{R} 0, 1$
$C \leftarrow \mathcal{E}(pk, m_0)$
Append C *to* CLIST
Returns C

For a given distinguishing algorithm D, We define the advantage

$$\mathbf{Adv}^{PA2I}_{\mathcal{A},\mathcal{A}^*,D} = |\Pr[D(x_{Real}) = 1] - \Pr[D(x_{Fake}) = 1]|$$

where x_{Real} is the output of $\mathcal{A}$ in the Real game and x_{Fake} is the output of $\mathcal{A}$ in the Fake game.

Essentially, the difference between the definitions of PA2 and PA2I is that we restrict to a plaintext creator $\mathcal{P}_I$ which takes two messages, chooses one of them at random and returns it. Note that this is sufficient to prove the theorem IND-CPA + PA2I $\implies$ IND-CCA2.

Definition 6 (PA+). *For any plaintext awareness definition PA (PA1, PA2I, PA2), we define a new condition PA+ (PA1+, PA2I+, PA2+) by adding a randomness oracle, which takes no input and returns a random bit. The plaintext extractor is altered so that it takes a list* RLIST *of all such bits queried so far as one of its inputs, i.e. $\mathcal{A}^*(pk, C, R[\mathcal{A}], \text{RLIST}, \text{CLIST}, state)$. Note that any such PA+ definition implies the corresponding PA definition, since an adversary may simply not use the randomness oracle.*

3 Theoretical results about Plaintext Awareness

In this section of the paper, we present our theoretical results about plaintext awareness.

3.1 Connection between PA2 and PA2+

Clearly, a scheme which is PA2+ must necessarily be PA2, since an adversary may simply not use its randomness oracle, but the converse is not obviously true. We now show that it is true for a sufficiently randomized encryption scheme, since an adversary may use randomness inherant in a ciphertext generated by the encryption oracle to simulate a randomness oracle. This in turn implies the scheme is PA1+, thus giving a formal proof of a result conjectured in [6].

Definition 7 (γ-Uniformity). *An encryption scheme is γ-uniform if for all messages m and ciphertexts C, $\Pr[\mathcal{E}(m) = C] \leq \gamma$, where the probability is taken over the choice of random coins used by the $\mathcal{E}$ algorithm.*

Definition 8 (Universal$_2$ Hash Family). *A family* $\mathbf{H} = (H, K, A, B)$ *of functions* $(H_k)_{k \in K}$ *where each* H_k *maps* A *to* B *is universal*$_2$ *if for all* $x \neq y$ *in* A, $\Pr[H_k(x) = H_k(y) | k \xleftarrow{R} K] \leq 1/|B|$.

Theorem 1. *Suppose a public key encryption scheme* Π *is* γ*-uniform and PA2 (resp. PA2I). Then it is PA2+ (resp PA2I+).*

We will use a universal$_2$ function family $\mathbf{H} = (H_k)_{k \in K}$ where H_k is a function from $\{0,1\}^* \to \{0,1\}$ for all $k \in K$. For simplicity, we will assume $K = \{0,1\}^n$. Such families are known to exist without any computational assumptions [11].

Lemma 1. *Fix a message* $m \in \mathcal{M}$*, a public key* pk*. Let* $k \xleftarrow{R} \{0,1\}^n$*. Then*

$$|\Pr[H_k(\mathcal{E}(pk, m)) = 1] - \frac{1}{2}| \leq \sqrt{2\gamma}\,,$$

where the probability is taken over the choice of k *and the random coins used by* $\mathcal{E}$*.*

Proof. We will use the Leftover Hash Lemma, as proved in Theorem 6.21 of [9]. Let $\mathcal{C}$ be the distribution of $C \leftarrow \mathcal{E}(pk, m)$. Let

$$\kappa(\mathcal{C}) = \sum_{v \in \{0,1\}^*} \Pr[C = v]^2\,.$$

κ is known as the collision probability, because it is equal to the probability that two random values sampled independently from $\mathcal{C}$ are equal, i.e. they collide.

Note that by the γ-uniformity property of the scheme,

$$\max_{v \in \{0,1\}^*} \Pr[C = v] \leq \gamma$$

So

$$\begin{aligned}\kappa(\mathcal{C}) &\leq \sum_{v \in \{0,1\}^*} \Pr[C = v]\gamma \\ &= \gamma \sum_{v \in \{0,1\}^*} \Pr[C = v] \\ &= \gamma\end{aligned}$$

then the Leftover Hash Lemma states that the statistical distance

$$\Delta[(H_k(x), k), (z, k)] \leq \sqrt{2\kappa}/2\,,$$

where $z \xleftarrow{\mathrm{R}} \{0,1\}$. Since $\kappa \leq \gamma$, this implies

$$\frac{1}{2}|\Pr[H_k(x) = 1] - 1/2| \leq \sqrt{2\gamma}/2\,,$$

so $|\Pr[H_k(x) = 1] - 1/2] \leq \sqrt{2\gamma}$.

Proof of Theorem 1. Let **H** be as above.

Let $\mathcal{A}$ be a PA2+ ciphertext creator. We construct a PA2 ciphertext creator $\mathcal{B}$ as follows: $\mathcal{B}$ takes input pk. We designate the first q_r n-bit chunks of the random tape of $\mathcal{B}$ as $(k_1, \ldots, k_{q_r})$ and the rest $R[\mathcal{A}]$. $\mathcal{B}$ runs $\mathcal{A}(pk; R[\mathcal{A}])$. It answers $\mathcal{A}$'s encryption and decryption queries by passing them to its own oracle and returning the result. To answer the i^{th} randomness query, it queries the encrypt oracle on input 0 and receives a ciphertext C. It then computes $b_i \leftarrow H_{k_i}(C)$ and returns b_i.

Since $\mathcal{B}$ is a valid $PA2$ ciphertext creator, there is a plaintext extractor $\mathcal{B}^*$. We use $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$ for $\mathcal{A}$.

$\mathcal{A}^*$ takes input $(pk, C, R[\mathcal{A}], \textsc{List}, \textsc{Rlist}, state)$. and runs as follows: If $state$ is the empty string, it chooses $(k_1, \ldots, k_{q_r}) \xleftarrow{R} (\{0,1\}^n)^{q_r}$. Otherwise parses $state$ as $((k_1, \ldots, k_{q_r}), state_{\mathcal{B}}, n_R, \textsc{List}')$. Let $n'_R = |\textsc{Rlist}|$ If $n'_R > n_R$, then there are new random bits b_i in Rlist, so for each random bit b_i after the n_R^{th}, it and generates a ciphertext C_i by running $\mathcal{E}(pk, \mathcal{P}(0))$ repeatedly until it finds one such that $H_{k_i}(C_i) = b_i$. We limit the number of ciphertexts it generates to λ, where λ is the security parameter. For each trial, $|\Pr[H_{k_i}(C_i) = b_i] - \frac{1}{2}| \leq \sqrt{2\gamma}$ by lemma 1. Thus $\mathcal{A}^*$ fails to find such a C_i with proability at most $(\frac{1}{2} + \sqrt{2\gamma})^\lambda$, which is negligible as a function of the security parameter λ as required. (This is because $\gamma(\lambda)$ is negligible, so there is a constant c such that $\gamma(\lambda) \leq 1/32$ for all $\lambda > c$. Thus $\sqrt{2\gamma(\lambda)} \leq 1/4$ for $\lambda > c$. Thus $(\frac{1}{2} + \sqrt{2\gamma})^\lambda \leq (\frac{3}{4})^\lambda$ which is negligible). This ensures that the running time is polynomial, but it fails with negligible probability. It then appends $(C_{n_R+1}, \ldots, C_{n'_R}, C)$ to $\textsc{List}'$ and computes $(m, state_{\mathcal{B}}) \leftarrow \mathcal{B}^*(pk, C, k_1 || \ldots || k_{q_d} || R[\mathcal{A}], \textsc{List}', state_{\mathcal{B}})$. Finally, it returns $(m, ((k_1, \ldots, k_{q_d}), state_{\mathcal{B}, n'_R, \textsc{List}'}))$.

We now show that $\mathcal{A}^*$ is a valid plaintext extractor for $\mathcal{A}$, i.e. the output $x \leftarrow \mathcal{A}^{\mathcal{O}}(pk)$ is computationally indistinguishable in the real and fake game. To that end, we fix a plaintext creator $\mathcal{P}$ and a distinguishing algorithm D and let S_i be the event that $D(x) = 1$ in game i.

Game 0: Let game 0 be the real game for $\mathcal{A}$, i.e:

$(pk, sk) \leftarrow \mathcal{G}(1^k)$
$x \leftarrow \mathcal{A}^O(pk)$
$d \leftarrow D(x)$

$\mathcal{A}$ has access to encryption, decryption and randomness oracles. The encryption oracle takes input s, computes $(m, state_{\mathcal{P}}) \leftarrow \mathcal{P}(s, state_{\mathcal{P}})$, computes $C \leftarrow \mathcal{E}(pk, m)$, adds C to List, and returns C. The decryption oracle $\mathcal{D}$ takes a ciphertext C not in List as input and returns $m \leftarrow \mathcal{D}(sk, C)$. The randomness oracle $\mathcal{R}$ takes no input and returns a random bit b.

Game 1: We modify the randomness oracle so that on the i^{th} query it chooses $b_i \leftarrow H_{k_i}(\mathcal{E}(pk, \mathcal{P}(0)))$.

$|\Pr[b_i = 1] - \frac{1}{2}| \leq \sqrt{2\gamma}$ The statistical distance $\Delta[(k_i, b_i), (k_i, b) \leq \sqrt{2\gamma}$ (where $b \xleftarrow{R} \{0,1\}$) by lemma 1. We consider the event $S_{1,j}$ that $\{D(x) = 1 | x \xleftarrow{R}$

$\mathcal{A}^O(pk)$ in a game where the first j randomness queries are handled as in game 1 and queries after the j^{th} are handled as in game 0. Then

$$\frac{1}{2}|\Pr[S_{1,j+1}] - \Pr[S_{1,j}]| \leq \sqrt{2\gamma}\,,$$

since $\Delta[b, b_{j+1}] \leq \sqrt{2\gamma}$. Thus by a hybrid argument,

$$|\Pr[S_1] - \Pr[S_0]| \leq 2q_R\sqrt{2\gamma}\,,$$

Game 2: We modify the randomness oracle so that it adds each ciphertext it generates to CLIST. Since $\mathcal{A}$ does not have access to CLIST, the view of $\mathcal{A}$ is identical in the two games unless it generates one of these ciphertext independently, and queries the $\mathcal{D}$ oracle. Since the ciphertexts in CLIST were generated properly, the chance that any given decryption query matches a particular ciphertext in the list is at most γ by the γ-uniformity property. Since there are q_R such ciphertexts and it makes at most q_d decryption queries,

$$|\Pr[S_2] - \Pr[S_1]| \leq q_R q_d \gamma$$

Game 3: We modify the decryption oracle so that it uses $\mathcal{A}^*$ to answer decryption queries. Game 3 exactly simulates the environment of $\mathcal{B}^*$, so if D is an arbitrary distinguishing algorithm for $\mathcal{B}$,

$$|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\mathrm{PA2}}_{\mathcal{B},\mathcal{B}^*,\mathcal{P},D}$$

Game 3 is the fake game for $\mathcal{A}$, so

$$\begin{aligned}\mathbf{Adv}^{\mathrm{PA2+}}_{\mathcal{A},\mathcal{A}^*,\mathcal{P},D} &= |S_3 - S_0| \\ &\leq \mathbf{Adv}^{\mathrm{PA2}}_{\mathcal{B},\mathcal{B}^*,\mathcal{P},D} + q_R q_e \gamma + q_R\sqrt{2\gamma}\end{aligned}$$

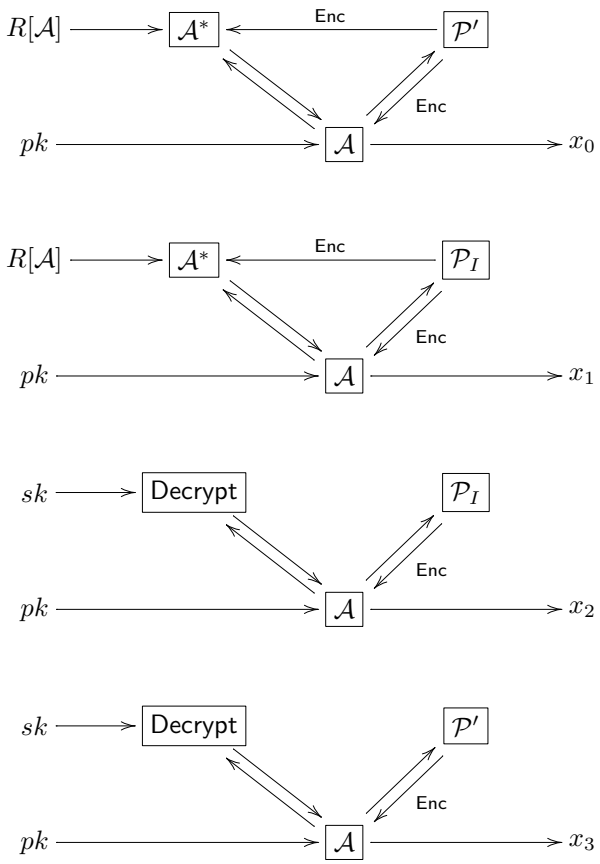
which is negligible as required.

3.2 Connection between PA2I and PA2

We note that PA2 trivially implies PA2I, as one may simply choose the plaintext creator to be $\mathcal{P}_I$ which takes two messages as input and returns one of them at random. We will now prove that for schemes which are IND-CPA, the reverse implication holds.

Theorem 2. *If an encryption scheme with finite message space $\mathcal{M} = \{0,1\}^\ell$ for some ℓ is IND-CPA and PA2I then it is PA2. Note that we could have equivalently chosen the message space to be $\{0,1\}^{<\ell}$, i.e. the set of bitstrings of length less than ℓ, as we can trivially map one set onto the other. Note also that ℓ may depend on the security parameter λ.*

Proof. Consider an arbitrary plaintext creator $\mathcal{P}'$. We first present 4 diagrams:



The diagrams show the games Game 0 to Game 3 respectively. We let x_i be the output of $\mathcal{A}$ in Game i. We now describe the games fully. We fix a distinguishing algorithm D and let S_i be the event that $D(x_i) = 1$.

Game 0: Let Game 0 be the fake game with plaintext creator $\mathcal{P}'$.

Game 1: We replace $\mathcal{P}'$ with the $\mathcal{P}_I$. Since $\mathcal{A}$ expects to be interacting with $\mathcal{P}'$, and will not explicitly format its queries as (m_0, m_1), we will define $\mathcal{P}_I$ so that it truncates or pads s with zeros to 2ℓ bits if necessary, and then splits the result into two ℓ bit messages, chooses one of them at random and returns it. The game then encrypts it and returns it to $\mathcal{A}$ and adds it to CLIST.

$\Pr[S_1] - \Pr[S_0]|$ is non-negligible, then we will construct an adversary $\mathcal{B}$ against the IND-CPA security of the scheme as follows:

Let q be an upper bound on the number of $\mathcal{E}$ queries made by $\mathcal{A}$. We consider a sequence of games $(1, 0), \ldots, (1, q)$ such that in game $(1, i)$ the first $q_e - i$ encryption queries are handled as in game 0 and all subsequent queries are handled as in game 1. The reason for doing it this way around is so that the state of $\mathcal{P}'$ is the same for the j^{th} query in all games for $j < q_e - i$. We let $x_{1,i}$ be the output of game $(1, i)$, and $S_{1,i}$ be the event that $D(x_{1,i}) = 1$.

Let

$$|\Pr(S_{1,i} = 1] - \Pr[S_{1,i+1} = 1]| = \epsilon_i\,.$$

We construct an IND-CPA adversary as follows:
$\mathcal{B}_1(pk)$

$R[\mathcal{A}] \xleftarrow{R} \{0,1\}^k$
$R[\mathcal{A}^*] \xleftarrow{R} \{0,1\}^k$
$R[\mathcal{P}'] \xleftarrow{R} \{0,1\}^k$
$\textsc{List} \leftarrow ()$
$state_{\mathcal{A}^*} \leftarrow \varepsilon$
$state_{\mathcal{P}} \leftarrow \varepsilon$
Run $\mathcal{A}(pk; R[\mathcal{A}])$ until i^{th} encryption query on input s_i.
$m_0 \leftarrow \mathcal{P}'(s_i, state_{\mathcal{P}})$
$m_1 \leftarrow \mathcal{P}_I(s_i)$
Return $(m_0, m_1, (pk, R[\mathcal{A}], R[\mathcal{A}^*], R[\mathcal{P}']))$

$\mathcal{B}_1$ answers $\mathcal{D}(C)$ queries as follows:

$(m, state_{\mathcal{A}^*}) \leftarrow \mathcal{A}^*(pk, C, R[\mathcal{A}], \textsc{List}, state_{\mathcal{A}^*}; R[\mathcal{A}^*])$
Return m

For the first $q_e - i$ encryption queries, (on input s) $\mathcal{B}_1$ answers as follows:

$(m, state_{\mathcal{P}}) \leftarrow \mathcal{P}'(s, state_{\mathcal{P}}; R[\mathcal{P}])$
$C \leftarrow \mathcal{E}(pk, m)$
Return m

$\mathcal{B}_2$ takes a challenge ciphertext C^* and $state = (pk, R[\mathcal{A}], R[\mathcal{A}^*], R[\mathcal{P}])$. It runs $\mathcal{A}$ again using the same randomness for each algorithm as before, so the exact state of $\mathcal{A}$ is restored to as it was when $\mathcal{A}_1$ terminated, and responds to the current encryption query with C^*. From that point on it responds to all encryption queries using $\mathcal{P}_I$ instead of $\mathcal{P}'$, but answers all other queries as before.

$\mathcal{B}_2(C^*, state)$
Parse $state$ as $(pk, R[\mathcal{A}], R[\mathcal{A}^*], R[\mathcal{P}']))$
$\textsc{List} \leftarrow ()$
$state_{\mathcal{A}^*} \leftarrow \varepsilon$
$state_{\mathcal{P}} \leftarrow \varepsilon$
$x \leftarrow \mathcal{A}(pk; R[\mathcal{A}])$
$b \leftarrow D(x)$
Return b

$\mathcal{B}$ responds to decryption queries and the first $q_e - i$ encryption queries as before. It responds to the i^{th} encryption query with C^*. It responds to all subsequent encryption queries as follows:

$m \leftarrow \mathcal{P}_I(s)$
$C \leftarrow \mathcal{E}(pk, m)$
Return m

If the challenger chooses bit message m_0, then $\mathcal{B}$ exactly simulates Game $(1, i)$. If it chooses m_1, then $\mathcal{B}$ exactly simulates Game $(1, i+1)$. By assumption, D distinguishes between the outputs of these games with advantage ϵ_i, so $\mathcal{B}$ has advantage ϵ_i against the IND-CPA security of the scheme, and $|\Pr[S_1] - \Pr[S_0]| \leq q_e \mathbf{Adv}_B^{\text{IND-CPA}}$

Game 2: We replace $\mathcal{A}^*$ with a real decryption oracle. By definition:

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}^{\text{PA2I}}_{\mathcal{A},\mathcal{A}^*,D}$$

Game 3: We replace $\mathcal{P}_I$ by $\mathcal{P}'$. We can prove that
$|\Pr[S_3 - \Pr[S_2]|$ is negligible by much the same argument, except that this time, we construct an IND-CCA2 adversary $\mathcal{C}$, which uses its own decryption oracle to answer decryption queries. We note that since $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is IND-CPA and PA2I, it is necessarily IND-CCA2 secure. The only difference is that in Game $(3, i)$ the first i encryption queries are handled as in game 3, using $\mathcal{P}'$, while subsequent queries are handled using P_I, again to ensure that the state of $\mathcal{P}'$ is the same for the j^{th} query in all games $j > i$, thus

$$|\Pr[D(x_3) = 0] - \Pr[D(x_2) = 0]| \leq q_e \mathbf{Adv}_C^{\text{IND-CCA}}$$

Since the scheme is PA2I and IND-CPA secure, there is an IND-CPA adversary $\mathcal{F}$ and tuple $(\mathcal{G}, \mathcal{G}^*, D')$ (where $\mathcal{G}$ is a PA2I ciphertext creator, $\mathcal{G}^*$ is a plaintext extractor and D' a distinguishing algorithm) such that $\mathbf{Adv}_{\mathcal{C}}^{\text{IND-CCA}} \leq \mathbf{Adv}_{\mathcal{F}}^{\text{IND-CPA}} + q_e \mathbf{Adv}^{\text{PA2I}}_{\mathcal{G},\mathcal{G}^*,D'}$. Thus the advantage

$$\begin{aligned}\mathbf{Adv}^{\text{PA2}}_{\mathcal{A},\mathcal{A}^*,\mathcal{P}',D} &\leq q_e \mathbf{Adv}_{\mathcal{B}}^{\text{IND-CPA}} + \mathbf{Adv}^{\text{PA2I}}_{\mathcal{A},\mathcal{A}^*,D} + q_e \mathbf{Adv}_{\mathcal{C}}^{\text{IND-CCA}} \\ &\leq q_e \mathbf{Adv}_{\mathcal{B}}^{\text{IND-CPA}} + \mathbf{Adv}^{\text{PA2I}}_{\mathcal{A},\mathcal{A}^*,D} + q_e(q_e \mathbf{Adv}_{\mathcal{F}}^{\text{IND-CPA}} + \mathbf{Adv}^{\text{PA2I}}_{\mathcal{G},\mathcal{G}^*,D'})\end{aligned}$$

is negligible as required.

3.3 Length hiding properties

In the above we restricted to a finite message space $\mathcal{M}$. We now show that this is necessary, not only for this result but to achieve PA2 in general.

Theorem 3. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. If Π is OW-CPA and has an infinite message space, then it is not PA2.*

Proof (Sketch Proof).

We will prove the equivalent condition that if $\mathcal{A}$ is PA2 and has an infinite message space, then it is not OW-CPA.

We use Teranishi and Ogata's result of [10] which states that if an encryption scheme is OW-CPA and PA2 then it is IND-CPA, and note that the proof actually shows that the scheme satisfies the stronger property that no adversary can win the IND-CPA game even if the restriction that $|m_0| = |m_1|$ is lifted. This is not possible if the message space is not finite, since the adversary may simply choose a one bit message m_0 and a message m_1 longer than the running time of $\mathcal{E}(pk, m_0)$. The bit b may then be determined simply by looking at the length of the ciphertext.

Briefly, Teranishi and Ogata's proof works by assuming that a scheme Π is PA2 but not IND-CPA. They break the OW property of the scheme by constructing a plaintext creator $\mathcal{P}$ which sends the ciphertext creator $\mathcal{A}$ a ciphertext for which it does not "know" the decryption. This is done by having $\mathcal{A}$ pass the public key pk to $\mathcal{P}$. $\mathcal{P}$ chooses a random message and encrypts it *itself* to get a ciphertext C. $\mathcal{A}$ then makes many encryption queries to $\mathcal{P}$ which allows $\mathcal{P}$ to send C to $\mathcal{A}$ one bit at a time using he fact that the scheme is not IND-CPA secure. We simply note that in particular, $\mathcal{P}$ may do this by sending short messages to represent a 0 and long messages to represent a 1.

$\mathcal{A}$ may then query its decryption oracle on this ciphertext, since it does not appear in CLIST. Any plaintext extractor for this scheme then allows us to break the OW-CPA property.

Proof (Full Proof).

We will show the equivalent condition that if $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2 and has the infinite message space $\mathcal{M} = \{0,1\}^*$, and prove that Π is not OW-CPA secure.

The length of a ciphertext is bounded by a polynomial $f(\lambda, |m|)$ in the security parameter λ and length of the corresponding plaintext. An upper bound for f is simply the running time of $\mathcal{E}$. Let $l_0 = 1$, $l_1 = f(\lambda, 1) + \lambda + 1$, and $l_2 = f(\lambda, l_1) + \lambda + 1$.

Let Encode be an algorithm which takes input $i \in \{0,1,2\}$ outputs a message $m \xleftarrow{R} \{0,1\}^{l_i}$. Let Decode be an algorithm which takes a ciphertext $C \leftarrow \mathcal{E}(pk, \mathsf{Encode}(i))$ returns a guess for i as follows:

```
Decode(C)
  If |C| ≤ f(λ, 1)
    Return 0
  Else if |C| ≤ f(λ, l_1)
    Return 1
  Else
    Return 2
```

If $i = 0$, Decode returns 0 with certainty, because $|C|$ is bounded by $f(\lambda, 1)$. If $i = 1$, it is possible that $|C| < f(\lambda, 1)$. However, there are at most $2^{f(\lambda,1)+1} - 1$

bitstrings of length $\leq f(\lambda, 1)$ and 2^{l_1} messages of length l_1. Since each bitstring is a valid encryption of at most one message, by a counting argument we see that

$$\begin{aligned}\Pr[\mathsf{Decode}(\mathcal{E}(pk, \mathsf{Encode}(1))) = 0] &= \Pr[|C| \leq f(\lambda, 1) \mid m \xleftarrow{\mathrm{R}} \{0,1\}^{l_1}; C \leftarrow \mathcal{E}(pk, m)] \\ &\leq \frac{2^{f(\lambda,1)+1} - 1}{2^{f(\lambda,1)+\lambda+1}} \\ &< 2^{-\lambda}\end{aligned}$$

Similarly, $\Pr[\mathsf{Decode}(\mathcal{E}(pk, \mathsf{Encode}(2))) \neq 2] \leq 2^{-\lambda}$. So for all i,

$$\Pr[\mathsf{Decode}(\mathcal{E}(pk, \mathsf{Encode}(i))) = i] \geq 1 - 2^{-\lambda}$$

We will construct a plaintext creator $\mathcal{P}$ and a ciphertext creator $\mathcal{A}$. $\mathcal{P}$ generates a random message m^* and a ciphertext $C^* \leftarrow \mathcal{E}(pk, m^*)$. It then uses Encode and Decode to send C^* to the ciphertext creator $\mathcal{A}$ one bit at a time. $\mathcal{P}$ uses the value 2 to indicate the end of message. Since $\mathcal{A}$ may fail to decode this correctly, it terminates after $f(\lambda, \ell)$ queries, to ensure the running time is polynomially bounied. $\mathcal{A}$ then obtains m by querying the decryption oracle on C^*. One final call to the encryption oracle allows $\mathcal{A}$ to check that the plaintext m is correct. Written out in full, $\mathcal{P}$ and $\mathcal{A}$ are as follows:

$\mathcal{P}(aux, state)$:
  If $state = \varepsilon$
    $pk = aux$
    $m^* \xleftarrow{\mathrm{R}} \{0,1\}^{\ell}$
    $C^* \leftarrow \mathcal{E}(pk, m^*)$
    Parse C^* as a string of bits $(b_0, \ldots, b_n)$
    $phase \leftarrow 0$
    $state \leftarrow (m^*, C^*, 1, phase)$
    Return $(\mathsf{Encode}(b_0), state)$
  Else
    Parse $state$ as $(m, C^*, i, phase)$
    If $phase = 0$
      Parse C^* as a string of bits $(b_0, \ldots, b_n)$
      If $i \leq n$
        Return $(\mathsf{Encode}(b_i), (m^*, C^*, i+1, 0))$
      Else
        Return $(\mathsf{Encode}(2), (m^*, C^*, i+1, 1))$
    Else
      If $aux = m^*$
        Return $(\mathsf{Encode}(1), \varepsilon)$
      Else
        Return $(\mathsf{Encode}(0), \varepsilon)$

$\mathcal{A}(pk)$:
  For i from 0 to $f(\lambda, \ell)$
    Call encryption oracle on pk to get ciphertext C_i
    $b_i \leftarrow \mathsf{Decode}(C_i)$
    If $b_{i-1} = 2$
      Break
  If $i \geq f(\lambda, \ell)$
    Return 0
  $C' \leftarrow b_0 || \ldots || b_{i-1}$
  Call decryption oracle on C' to get a message m
  Call encryption oracle on m to get ciphertext C
  $b \leftarrow \mathsf{Decode}(C)$
  Return b

Let S_{real} be the event that $\mathcal{A}$ returns 1 in the real game, and S_{fake} be the event that $\mathcal{A}$ returns 1 in the fake game. Decode correctly returns each bit b_i from the input C_i, and the 2 which signifies the end of string with probability at least $1 - 2^{-\lambda}$. Thus $\mathcal{A}$ decodes $C' = b_0 || \ldots || b_n = C^*$ correctly with probability at least $1 - (|C^*| + 1) \cdot 2^{-\lambda}$. Assuming $C' = C^*$, the decryption oracle correctly returns $m = m^*$. In this case, $\mathcal{P}$ returns $\mathsf{Encode}(1)$ in response to the final query, so $\mathcal{A}$ returns the bit 1 with probability $1 - 2^{-\lambda}$. Thus

$$\Pr[S_{real}] \geq 1 - (|C^*| + 2) \cdot 2^{-\lambda}.$$

On the other hand, in either game if the message $m \neq m^*$, $\mathcal{A}$ returns 0 with certainty, since $\Pr[\mathsf{Decode}(\mathcal{E}(pk, \mathsf{Encode}(0))) = 0] = 1$.

Since the encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2 plaintext aware, there exists a plaintext extractor $\mathcal{A}^*$ such that the output of $\mathcal{A}$ in the fake game is indistinguishable from the output of $\mathcal{A}$ in the real game. Since $\mathcal{A}$ returns a single bit, this implies $|\Pr[S_{real}] - \Pr[S_{fake}]| \leq \epsilon$, for some negligible ϵ. Thus $\Pr[S_{fake}] \geq 1 - (|C^*| + 2) \cdot 2^{-\lambda} - \epsilon$ This implies that $\mathcal{A}^*$ returns $m = m^*$ correctly with probability at least $1 - (|C^*| + 2) \cdot 2^{-\lambda} - \epsilon$.

We use $\mathcal{A}^*$ to construct an adversary $\mathcal{B}$ against the OW-CPA security of Π as follows:

$\mathcal{B}(pk, C^*)$:
  Parse C^* as a string of bits $(b_0, \ldots, b_n)$
  For $i = 0$ to n
    $C_i \leftarrow \mathcal{E}(pk, \mathsf{Encode}(b_i))$
  $C_{n+1} \leftarrow \mathcal{E}(pk, \mathsf{Encode}(2))$
  $\textsc{Clist} \leftarrow (C_0, \ldots, C_{n+1})$
  $R[\mathcal{A}] \leftarrow \varepsilon$
  $state_{\mathcal{A}^*} \leftarrow \varepsilon$
  $m \leftarrow \mathcal{A}^*(pk, C^*, R[\mathcal{A}], \textsc{Clist}, state_{\mathcal{A}^*})$
  Return m

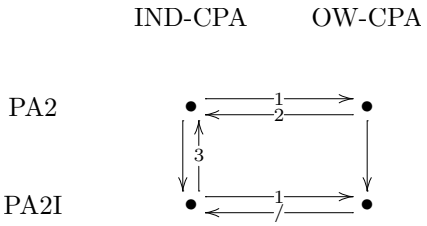
$\mathcal{B}$ succedes in breaking the OW-CPA security of Π if $\mathsf{Decode}(C_i) = b_i$ for all $0 \leq i \leq n$ and $\mathcal{A}^*$ returns the correct decryption of C^*. Since $\mathcal{B}$ simulates the environment of the fake game exactly,

$$\mathbf{Adv}_{\mathcal{B}}^{\text{OW-CPA}} \geq 1 - (|C^*| + 2) \cdot 2^{-\lambda} - \epsilon$$

which is non-negligible as required.

3.4 Relationship between IND-CPA, OW-CPA and PA2I

The diagram below illustrates relationships between IND-CPA, OW-CPA, PA2 and PA2I. The numbered and plain arrows show implications, while the slashed arrow shows a separation, which we will discuss later.



The downwards arrows in the diagram follow trivially, since PA2I is a weaker notion than PA2. The arrows numbered 1 follow trivially if the message space is super-polynomial sized in the security parameter, thus any scheme which is IND-CPA is also OW-CPA.

The arrow numbered 2 indicates that a scheme which is OW-CPA and PA2 must also be IND-CPA (and hence IND-CCA), as shown by Teranish and Ogata [10]. The arrow numbered 3 indicates that a scheme which is IND-CPA and PA2I is also PA2, as we showed earlier.

It does not appear to be the case that PA2I and OW-CPA is enough to show that a scheme is IND-CPA. To prove the separation, ideally we would show that if their is a scheme which is OW-CPA and PA2I but not PA2. Since it is even not known that OW-CPA secure schemes even exist without computational assumptions, it is necessary to at least assume that there is a scheme which is OW-CPA and PA2I. Unfortunately, to attain this result we have had to make the slightly stronger assumption that there is a scheme which is OW-CPA and PA2I+. As shown earlier, this is implied if there is a γ-uniform encryption scheme which is PA2I and OW-CPA.

Theorem 4. *Assume there exists an encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ with message space $\mathcal{M} = \{0,1\}^\ell$ which is OW-CPA, IND-CPA, and PA2I+. Then there exists another encryption scheme $\Pi' = (\mathcal{G}, \mathcal{E}', \mathcal{D}')$ which is OW-CPA and PA2I+ but not IND-CPA.*

Proof. We let $F(m)$ denote the final bit of message m. We now describe a new encryption scheme $\Pi' = (\mathcal{G}, \mathcal{E}', \mathcal{D}')$, where the final bit of a message is leaked by the encryption algorithm. In full, $\mathcal{G}$ is as in Π and $\mathcal{E}'$ and $\mathcal{D}'$ are as follows:

$\mathcal{E}'(pk, m)$:
  $C' \leftarrow \mathcal{E}(pk, m)$
  $b \leftarrow F(m)$
  $C \leftarrow (C', b)$
  Return C

$\mathcal{D}'(sk, C)$:
Parse C as (C', b)
$m \leftarrow \mathcal{D}(sk, C')$
If $b = F(m)$:
  Return m
Else
  Return $\perp$

Clearly, Π' is OW-CPA, since if there is an adversary against the OW-CPA security of Π' with advantage ϵ, there is an adversary against Π with advantage $\epsilon/2$ which just guesses the final bit at random. It is also clear that Π' is not IND-CPA, since an adversary may simply choose messages $m_0 = 0^\ell$ and $m_1 = 0^{\ell-1}||1$.

We must now prove that Π' is PA2I. Let $\mathcal{A}$ be a PA2I ciphertext creator against Π'. We construct a PA2I+ ciphertext creator $\mathcal{B}$ against Π which works as follows:

$\mathcal{B}$ runs $\mathcal{A}(pk; R[\mathcal{B}])$ and handles queries as follows: When $\mathcal{A}$ makes an encrypt query on (m_0, m_1), $\mathcal{B}$ queries its own $\mathcal{E}$ oracle on (m_0, m_1) and receives a ciphertext C'. It then checks if $F(m_0) = F(m_1)$. If so, $\mathcal{B}$ then returns $(C', F(m_0))$ to $\mathcal{A}$. If not, $\mathcal{B}$ queries its randomness oracle to get a bit b, and returns (C', b).

When $\mathcal{A}$ makes a decryption query on $C = (C', b)$, $\mathcal{B}$ queries its own decryption oracle on C' to get a message m, and returns m if $F(m) = b$ or $\perp$ otherwise. Finally, when $\mathcal{A}$ outputs x and terminates, $\mathcal{B}$ does the same. By the PA2I+ property of Π there exists a plaintext extractor $\mathcal{B}^*$ for $\mathcal{B}$. We use $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$ for $\mathcal{A}$. $\mathcal{A}^*$ will respond to $\mathcal{B}^*$'s randomness queries using the real bit b_i it obtains from CLIST. In order to do this, it must work out which ciphertext queries would cause $\mathcal{B}$ to make a randomness query – i.e. those for which $F(m_0) \neq F(m_1)$. To do this, it simply runs $\mathcal{A}$.

$\mathcal{A}^*$ takes input $(pk, C, R[\mathcal{A}], \text{CLIST}, state)$ and runs as follows:

$\mathcal{A}^*(pk, C, R[\mathcal{A}], \text{CLIST}, state)$
  Parse $state$ as $(i, (m_1, \ldots, m_i), state_{\mathcal{B}})$
  Parse CLIST as $((C'_1, b_1), \ldots, (C_j, b_j))$
  $\text{CLIST}_{\mathcal{B}} \leftarrow (C_1, \ldots, C_j)$
  $x \leftarrow \mathcal{A}^{\mathcal{O}}(pk)$ (see below for oracle queries)
  $(m_{i+1}, state_{\mathcal{B}}) \leftarrow \mathcal{B}^*(pk, C, R[\mathcal{A}], \text{RLIST}, \text{CLIST}_{\mathcal{B}}, state_{\mathcal{B}})$
  Return $(m_{i+1}, (i, (m_1, \ldots, m_i, m_{i+1}), state_{\mathcal{B}}))$.

It answers $\mathcal{A}$'s l^{th} $\mathcal{D}$ query with m_l for $1 \leq l \leq i$. It answers the l^{th} $\mathcal{E}$ query on input (m_0, m_1) with (C_l, b_l), for $1 \leq l \leq j$. If $F(m_0) \neq F(m_1)$, it also appends b_l to an initially empty list RLIST.

We must now show that $\mathcal{A}^*$ is a valid plaintext extractor for $\mathcal{A}$. We do this by showing that $\mathcal{A}^*$ almost perfectly simulates the environment of $\mathcal{B}^*$, using a sequence of games. We fix a distinguishing algorithm D and let S_i be the event that $D(x)$ returns 1 in game i.

Game 0: Let Game 0 be the real game for $\mathcal{A}$.

Game 1: We let Game 1 be as Game 0 except that for each ciphertext (C_i, b_i) returned by the encryption oracle, the bit b_i is chosen in the same way that $\mathcal{B}$

does – i.e. if $F(m_0) = F(m_1)$ then the oracle chooses $b_i = F(m_0)$, otherwise b is chosen uniformly at random $\{0,1\}$ independently of the message that is encrypted.

Game 1 exactly simulates the real game for $\mathcal{B}$. We claim that $|\Pr[S_1] - \Pr[S_0]| \leq q_e \mathbf{Adv}_{\mathcal{C}}^{\text{IND-CCA}}$, for some adversary $\mathcal{C}$, since if the outputs of $\mathcal{A}$ are distinguishable in these two games, we can construct an adversary which distinguishes ciphertexts.

Game 2: Let Game 2 be the same as Game 1, except that $\mathcal{A}$'s $\mathcal{D}$ queries are handled by $\mathcal{A}^*$. We note that Game 2 exactly simulates the fake game for $\mathcal{B}$, and in particular, the view of $\mathcal{B}^*$ is exactly as it would be in the fake game for $\mathcal{B}$. Thus by the PA2I+ property of Π, $|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}_{\mathcal{B},\mathcal{B}^*,D}^{\text{PA2I+}}$.

Game 3: Let Game 3 be as Game 2, except with the original behaviour of $\mathcal{E}$ restored, i.e. the final bit of the ciphertext is the final bit of the message.

Game 3 is identical to the fake game for $\mathcal{A}$. Furthermore, $|\Pr[S_3] - \Pr[S_2]| \leq q_e \mathbf{Adv}_{\mathcal{F}}^{\text{IND-CPA}}$, for some adversary $\mathcal{F}$, since if the outputs of $\mathcal{A}$ are distinguishable in these two games, we can construct an adversary against the IND-CPA security of Π.

Putting it all together, we see that

$$\begin{aligned} |\Pr[S_3] - \Pr[S_0]| &\leq |\Pr[S_1] - \Pr[S_0]| + \ldots + |\Pr[S_3] - \Pr[S_2]| \\ &= q_e \mathbf{Adv}_{\mathcal{C}}^{\text{IND-CCA}} + \mathbf{Adv}_{\mathcal{B},\mathcal{B}^*,D}^{\text{PA2I+}} + q_e \mathbf{Adv}_{\mathcal{F}}^{\text{IND-CPA}} \end{aligned}$$

which is negligible as required.

4 Plaintext awareness of Kurosawa-Desmedt

4.1 Definitions

In this section we recall the definitions and theorems about encryption schemes based on universal hash proof systems [5, 7]. Let X, S and Π be finite non-empty sets and let L be a non-empty subset of X. Let $(H_k)_{k \in K}$ be a collection of functions indexed by K such that $H_k : X \to \Pi$, and let $\alpha : K \to S$ be a polynomial-time computable function. Then the tuple $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ is a projective hash family if for every $k \in K$, the value $s = \alpha(k) \in S$ uniquely determines the action of H_k on L.

Definition 9. *A projective hash family* $\mathbf{H}$ *is is* ϵ*-universal*$_2$ *if for all* $x \in X \setminus L$*, for all* $x^* \in X \setminus \{x\}$*, for all* $\pi, \pi^* \in \Pi$*, for all* $s \in S$ *and for* k *chosen uniformly at random in* K*:*

$$\Pr[H_k(x) = \pi | H_k(x^*) = \pi^* \wedge \alpha(k) = s] \leq \epsilon$$

$\mathbf{H}$ *is* ϵ*-smooth if*

$$\max_{Z' \subset Z} |\Pr[(x, s, \pi') \in Z'] - \Pr[(x, s, \pi) \in Z']| \leq \epsilon$$

where $Z = X \times S \times \Pi$*,* $k \xleftarrow{R} K$*,* $s \leftarrow \alpha(k)$*,* $x \xleftarrow{R} X \setminus L$*,* $\pi \leftarrow H_k(x)$ *and* $\pi' \xleftarrow{R} \Pi$*.*

For our purposes, a subset membership problem $\mathbf{M}$ is defined by a probabilistic, polynomial-time *instance sampling algorithm* ISA, where $\Lambda[X, L, W, R] \xleftarrow{R} \mathsf{ISA}(1^\ell)$ with W and X non-empty sets, $R \subset X \times W$ a relation, and $L = \{x \in X : \exists w \in W \text{ such that } (x, w) \in R\}$, a proper, non-empty subset of X. We will assume that there is an efficient sampling algorithm that can compute (x, w) where $x \xleftarrow{R} L$ and $w \in W$ is chosen at random such that $(x, w) \in R$. We will abuse notation slightly, and write this as $(x, w) \xleftarrow{R} R$.

Definition 10. *A subset membership problem is* hard *if for any probabilistic, polynomial-time algorithm $\mathcal{A}$ and random instance $\Lambda[X, L, W, R] \leftarrow \mathsf{ISA}(1^\ell)$, the advantage of $\mathcal{A}$, given by:*

$$|\Pr[1 \leftarrow \mathcal{A}(\Lambda, x) \mid (x, w) \xleftarrow{R} R] - \Pr[1 \leftarrow \mathcal{A}(\Lambda, x) \mid x \xleftarrow{R} X \setminus L]|$$

is negligible.

A hash proof system (HPS) $\mathbf{P}$ for a subset membership problem $\mathbf{M}$ associates a projective hash family $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ to each instance $\Lambda[X, L, W, R]$ of $\mathbf{M}$. The HPS must also provide the following polynomial-time algorithms:

- $\mathsf{Private}_{\mathbf{H}}$ takes input (k, x) where $k \in K$ and $x \in X$ and returns $H_k(x)$.
- $\mathsf{Public}_{\mathbf{H}}$ takes input (s, x, w) where $s = \alpha(k)$ and $(x, w) \in R$, and returns $H_k(x)$

Note that the output of Public is well defined, since $\alpha(k)$ determines the function H_k on L, even though it may not determine the action of H_k on $X \setminus L$.

A more efficient version of the Cramer-Shoup encryption scheme [4] was developed by Kurosawa and Desmedt [7]. This scheme uses a symmetric encryption scheme (sometimes called a DEM, or data encapsulation mechanism) to encrypt the message with a key derived from the hash proof system via a hash function $\text{Hash} : \Pi \to \{0,1\}^\ell$.

A DEM is a pair of deterministic polynomial-time algorithms $(\mathsf{Enc}, \mathsf{Dec})$ where $\mathsf{Enc} : \{0,1\}^\ell \times \mathcal{M} \to \mathcal{C}$ and $\mathsf{Dec} : \{0,1\}^\ell \times \mathcal{C} \to \mathcal{M}$. We require that for all $\kappa \in \{0,1\}^\ell$ and $m \in \mathcal{M}$, $\mathsf{Dec}(\kappa, \mathsf{Enc}(\kappa, m)) = m$. We will require that the DEM is ϵ-rejection secure:

Definition 11 (Rejection-secure). *A DEM $(\mathsf{Enc}, \mathsf{Dec})$ is ϵ-rejection secure if for all $\chi \in \{0,1\}^*$,*

$$\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp \mid \kappa \xleftarrow{R} \{0,1\}^\ell] \leq \epsilon\,.$$

We will use a δ-smooth hash function to derive keys for an ϵ-rejection secure DEM.

Definition 12 (Smooth hash function). *A hash function $\text{Hash} : A \to B$ is δ-smooth if $\Delta[\text{Hash}(a), b] \leq \delta$, where $b \xleftarrow{R} B$ and $a \xleftarrow{R} A$*

We will now show that the resulting combination is $\epsilon + 2\delta$-rejection secure.

Lemma 2. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be an* ϵ*-rejection secure DEM which uses keys of length* ℓ *and* $\mathrm{Hash} : A \to \{0,1\}^\ell$ *be a* δ*-smooth hash function. Then for all* $\chi \in \{0,1\}^*$

$$\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp \mid a \xleftarrow{R} A; \kappa \leftarrow \mathrm{Hash}(a)] \leq \epsilon + 2\delta$$

Proof. Let $a \xleftarrow{R} A, \kappa \leftarrow \mathrm{Hash}(a)$ and let $\kappa' \xleftarrow{R} \{0,1\}^\ell$. Fix $\chi \in 0, 1^*$. Then by the δ-smoothness of Hash, $\Delta[\kappa, \kappa'] \leq \delta$, which implies that $\Delta[\mathsf{Dec}(\kappa, \chi), \mathsf{Dec}(\kappa', \chi)] \leq \delta$. So $\frac{1}{2}|\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp] - \Pr[\mathsf{Dec}(\kappa', \chi) \neq \perp]| \leq \delta$, or equivalently, $|\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp] - \Pr[\mathsf{Dec}(\kappa', \chi) \neq \perp]| \leq 2\delta$

But $\Pr[\mathsf{Dec}(\kappa', \chi) \neq \perp] \leq \epsilon$ by the ϵ-rejection secureness of the DEM. So

$$\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp \mid a \xleftarrow{R} A; \kappa \leftarrow \mathrm{Hash}(a)] \leq \epsilon + 2\delta$$

as required.

Definition 13 (Kurosowa-Desmedt). *Let* $\mathbf{M}$ *be a subset membership problem and let* $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ *be a hash proof system for* $\mathbf{M}$*. The Kurosawa-Desmedt universal hash proof encryption scheme is then defined as follows:*

$\mathcal{G}(\Lambda)$
- $sk \xleftarrow{R} K$
- $pk \leftarrow \alpha(sk)$
- *Return* (pk, sk)

$\mathcal{E}(pk, m)$
- $(x, w) \xleftarrow{R} R$
- $\pi \leftarrow \mathsf{Public}_{\mathbf{H}}(pk, x, w)$
- $\kappa \leftarrow \mathrm{Hash}(\pi)$
- $\chi \leftarrow \mathsf{Enc}(\kappa, m)$
- $C \leftarrow (x, \chi)$
- *Return* C

$\mathcal{D}(sk, C)$
- $\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
- $\kappa \leftarrow \mathrm{Hash}(\pi)$
- $m \leftarrow \mathsf{Dec}(\kappa, \chi)$
- *Return* m

Theorem 5 (Kurosawa-Desmedt). *If* $\mathbf{M}$ *is a hard subset membership problem,* $\mathbf{H}$ *is* $1/\Pi$*-universal*$_2$*,* $(\mathsf{Enc}, \mathsf{Dec})$ *is a DEM which is both one-time IND-CCA secure and* ϵ*-rejection secure, and if* Hash *is a* δ*-smooth hash function, then for negligible* ϵ *and* δ *the resulting scheme is IND-CCA2 secure.*

4.2 The Generalised Kurosawa-Desmedt Encryption Schemes Are Plaintext Aware

The classic example of a universal hash proof system is the basis of the Cramer-Shoup encryption scheme [5]. This was recently shown to plaintext aware under the non-standard DHK assumption [6] (and some standard assumptions). It is therefore natural to ask whether other encryption schemes based on other universal hash proof systems are plaintext aware. In this section, we show that the Kurosawa-Desmedt based on any universal hash proof system is plaintext aware if a generalisation of the DHK assumption, the *subset witness knowledge* (SWK) assumption holds. Loosely speaking, the subset witness knowledge assumption states that it is impossible to output an element $x \in L$ without also computing a witness $w \in W$ such that $(x, w) \in R$.

Definition 14 (Subset Witness Knowledge Assumption). *We first describe the SWK game for an adversary $\mathcal{A}$ and extractor $\mathcal{A}^*$. First, the challenger generates an instance $\Lambda[X, L, W, R] \leftarrow \mathsf{ISA}(1^\ell)$ of the a hard problem $\mathbf{M}$, and runs $\mathcal{A}^{\mathcal{O}}(\Lambda)$. $\mathcal{A}$ has access to an oracle $\mathcal{O}$ which takes an element $x \in X$ as input and runs as follows:*

$(w, state) \leftarrow \mathcal{A}^*(x, state, R[\mathcal{A}])$
Returns w

In the above, state is a state variable, initially set to the empty string ε.

A subset membership problem $\mathbf{M}$ satisfies the SWK assumption if for every probabilistic, polynomial-time algorithm $\mathcal{A}$, there exists a probabilistic, polynomial-time extractor $\mathcal{A}^*$ such that for all queries x, the probability

$$\Pr[(x, w) \notin R \mid x \in L\wedge]$$

is negligible as a function of ℓ. We stress that the interaction between $\mathcal{A}$ and $\mathcal{A}^*$ may continue for multiple "rounds" and that $\mathcal{A}^*$ must correctly find a witness each time. We also note that if $\mathcal{A}$ outputs $x \notin L$ then the response of $\mathcal{A}^*$ may be arbitrary.

We also require that the sets X, Π and $\hat{\Pi}$ are simulatable.

Definition 15 (Simulatable Set). *A set S is simulatable if there exists polynomial-time algorithms (f, f^{-1}) such that*

- *f is a deterministic algorithm which takes input $r \in \{0,1\}^n$ and outputs an element $s \in S$.*
- *f^{-1} is a probabilistic algorithm which takes input $s \in S$ and outputs a string $r \in \{0,1\}^n$*
- *For all $s \in S$, $f(f^{-1}(s)) = s$.*
- *For any probabilistic, polynomial-time algorithm $\mathcal{A}$,*

$$|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_1}(1^\ell)] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_2}(1^\ell)]|$$

  is negligible, where $\mathcal{O}_1$ takes no input and returns $r \xleftarrow{R} \{0,1\}^n$, and $\mathcal{O}_2$ takes no input and returns $f^{-1}(f(r))$ where $r \xleftarrow{R} \{0,1\}^n$.
- *For any probabilistic, polynomial-time algorithm $\mathcal{A}$,*

$$|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_3}(1^\ell)] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_4}(1^\ell)]|$$

  is negligible, where $\mathcal{O}_3$ takes no input, generates $r \xleftarrow{R} \{0,1\}^n$ and returns $f(r)$, and $\mathcal{O}_4$ takes no input and returns $s \xleftarrow{R} S$.

We extend the notion of a simulatable set to that of a simulatable symmetric encryption scheme. A symmetric encryption scheme is simulatable if the ciphertext space of the encryption scheme is simulatable as a set and for any probabilistic, polynomial-time algorithm $\mathcal{A}$,

$$|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_3, \mathsf{Dec}(\kappa,\cdot)}(1^\ell) \mid \kappa \xleftarrow{R} \{0,1\}^\ell] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_4, \mathsf{Dec}(\kappa,\cdot)}(1^\ell) \mid \kappa \xleftarrow{R} \{0,1\}^\ell]|$$

is negligible, where $\mathcal{O}_3$ takes a message m as input, generates $r \xleftarrow{R} \{0,1\}^n$ and returns $f(r)$, $\mathcal{O}_4$ takes a message m as input, computes $C \leftarrow \mathsf{Enc}(\kappa, m)$, appends C to CLIST and returns C and the Dec oracle takes a ciphertext $C \notin$ CLIST and returns $\mathsf{Dec}(\kappa, C)$.

We also require the notion of simulatable public-key encryption schemes. We say a public-key encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is simulatable if there exist polynomial-time algorithms f, f^{-1} with the following properties:

- f is a deterministic algorithm which takes input (pk, r) where pk is a public key and $r \in \{0,1\}^n$ and outputs an element $C \in \mathcal{C}$, the ciphertext space. Again, we will call f the simulation function.
- f^{-1} is a probabilistic algorithm which takes input (pk, C) where pk is a public key and $C \in \mathcal{C}$ and outputs a string $r \in \{0,1\}^n$. We will typically write f and f^{-1} without the public key inputs.
- For all $C \in \mathcal{C}$, $f(f^{-1}(C)) = C$.
- For any probabilistic, polynomial-time algorithm $\mathcal{A}$, and for a randomly generated keypair $(pk, sk) \leftarrow \mathcal{G}(1^\ell)$
$$|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_1}(1^\ell, pk)] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_2}(1^\ell, pk)]|$$
is negligible, where $\mathcal{O}_1$ takes no input and returns $r \xleftarrow{R} \{0,1\}^n$, and $\mathcal{O}_2$ returns $f^{-1}(f(r))$ where $r \xleftarrow{R} \{0,1\}^n$.
- For any probabilistic, polynomial-time algorithm $\mathcal{A}$, and for a randomly generated keypair $(pk, sk) \leftarrow \mathcal{G}(1^\ell)$
$$|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{E},\mathcal{D}}(1^\ell, pk)] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{FE},\mathcal{D}}(1^\ell, pl)]|$$
is negligible, where the decryption oracle $\mathcal{D}$ takes a ciphertext C and returns $m \leftarrow \mathcal{D}(sk, C)$, the encryption oracle $\mathcal{E}$ takes a message m and returns $\mathcal{E}(pk, m)$ and the $\mathcal{FE}$ oracle takes a message m, generates a random string $r \xleftarrow{R} \{0,1\}^n$ and returns $f(r)$. In either situation, $\mathcal{A}$ may not query the $\mathcal{D}$ oracle on a ciphertext returned by the relevant encryption oracle.

It has been noted by Stam that a simulatable encryption scheme is necessarily IND-CCA2 secure.

Theorem 6. *Suppose* $\mathbf{M}$ *is a hard membership problem for which the subset witness membership problems is hard, and that* $\mathbf{H}$ *is a hash proof system for* $\mathbf{M}$ *as in the description of the Kurosawa-Desmedt encryption scheme. Suppose that* S *and* X *are simulatable sets, and that* $\mathbf{H}$ *is* $1/|\Pi|$*-universal*$_2$*. Suppose that the hash function* Hash *is* δ*-smooth and that the symmetric encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *is both IND-CCA2 secure,* ϵ*-rejection secure and simulatable. Then the Kurosawa-Desmedt encryption scheme is PA2 plaintext aware.*

We use the proof strategy proposed by Dent to prove the Cramer-Shoup encryption scheme is plaintext aware [6], namely, we first prove that the scheme is PA1+ plaintext aware, and that it is simulatable. We then use the following theorem proved in [6]:

Theorem 7 (Dent). *If a public-key encryption scheme is PA1+ plaintext aware and simulatable, then it is PA2 plaintext aware.*

4.3 Kurosawa-Desmedt is PA1+

Theorem 8. *Suppose that* $\mathbf{H}$ *is strongly universal*$_2$*, that the symmetric encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *is* ϵ*-rejection secure, and that* S *is simulatable, with simulation function* f*. Assume also that the SWK assumption holds for the underlying subset membership problem* $\mathbf{M}$*. Then the generalised Kurosawa-Desmedt encryption scheme is PA1+.*

Proof. Let $\mathcal{A}$ be a PA1+ ciphertext creator which makes at most q randomness queries. We will construct a plaintext extractor $\mathcal{A}^*$ for $\mathcal{A}$ by constructing an SWK adversary $\mathcal{B}$ and using the existence of an SWK extractor $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$.

$\mathcal{B}$ takes input $\Lambda[X, L, W, R]$ and works as follows:

- Let r be the first n-bit block of the random tape of $\mathcal{B}$. Let $R_1, \ldots, R_q$ be the next q bits, i.e. write $R[\mathcal{B}]$ in the form $r||R_1||\ldots||R_q||R_A$ for some R_A.
- $s \leftarrow f(r)$
- $pk \leftarrow (\Lambda[X, L, W, R], s)$
- Run $\mathcal{A}(1^\ell, pk)$ with random coins R_A

$\mathcal{B}$ responds to the i^{th} randomness query made by $\mathcal{A}$ by returning R_i. To respond to $\mathcal{A}$'s decryption queries for a ciphertext $C = (x, \chi)$, $\mathcal{B}$ obtains a witness w by calling its SWK oracle on x and runs as follows:

If $(x, w) \notin R$
  Return $\perp$
$\pi \leftarrow \mathsf{Public}(s, x, w)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

Assume for the moment that (x, χ) is a valid ciphertext. Then $x \in L$. If the SWK oracle returns a valid witness, this always gives the correct decryption. On the other hand, if $x \notin L$, $\mathcal{B}$ will certainly reject, since there is no valid witness. However, in this case the strongly universal$_2$ property of $\mathbf{H}$ implies that $\pi = H_k(x)$ is uniformly distributed on Π. Thus by lemma 2 this ciphertext would be rejected by the real decryption oracle with probability at least $1-(\epsilon+2\delta)$. Thus $\mathcal{B}$'s responses to the decryption queries are computationally indistinguishable from those in the real game.

Since $\mathcal{B}$ is a valid SWK adversary, by the SWK assumption there is an extractor $\mathcal{B}^*$ which correctly responds to queries with probability $1 - \nu$ where ν is negligible. We use $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$:

To respond to the j-th decryption query (x_j, χ_j), $\mathcal{A}^*$ takes input a public key $pk = (\Lambda[X, L, W, R], s)$, the ciphertext (x_j, χ_j), the random coins $R[\mathcal{A}]$ of $\mathcal{A}$, a list of random bits $(R_1, \ldots, R_i)$ where i is the number of randomness queries made by $\mathcal{A}$ so far, and some state information in the form $(x_1, \ldots, x_{j-1})$ and runs as follows:

$R_{i+1}, \ldots, R_q \stackrel{R}{\leftarrow} \{0,1\}$
$r \leftarrow f^{-1}(s)$
$R_B = r||R_1|| \ldots ||R_i||R_{i+1}|| \ldots ||R_q||R[\mathcal{A}]$
$state \leftarrow \varepsilon$
For t from 1 to $j+1$:
  $(w_t, state) \leftarrow \mathcal{B}^*(\Lambda[X, L, W, R], x_t, R_B, state)$
Decrypt the ciphertext using w in the same way that $\mathcal{B}$ does as described above
Return $(m, (x_, \ldots, x_{j-1}, x_j))$

It is important to exactly simulate the environment of the real SWK game, since we only know the behaviour of $\mathcal{B}^*$ in that situation. A potential problem occurs when we consider the random tape of $\mathcal{B}$, since $\mathcal{B}^*$ expects to get all the randomness used by $\mathcal{B}$ as input, while in the PA1+ game, $\mathcal{A}$ is allowed to make randomness queries, the values of which are not known to $\mathcal{A}^*$ before the queries are made. In order to use $\mathcal{B}^*$, we provide it with all the correct random bits we know of, and make up random values using $\mathcal{A}^*$'s own random tape for the random bits that have not yet been requested. Since the distribution of these values is identical to the distribution of the "real" random values returned by the randomness oracle, the simulated environment is correct for the first query. If $\mathcal{A}$ requests more random bits and then makes a second decryption query, and $\mathcal{A}^*$ just called $\mathcal{B}^*$ on the new query x, $\mathcal{B}^*$ would be able to check that the random bits it received as input were the same as the previous time it was called by simply storing the input it received the first time in its state variable and comparing those values with the ones it received as input the second time it is called.

To avoid the problem, $\mathcal{A}^*$ re-initialises $\mathcal{B}^*$ with an empty state and calls it again on every previous query it has made to $\mathcal{B}^*$ before, but with all of the "correct" random bits it has available. The witnesses generated may be different to ones generated on previous occasions, but as $\mathcal{A}$ only receives the decryption m in response to its query, and m is correctly decrypted whichever witness is used, the witnesses are independent of $\mathcal{A}$'s view. This ensures that the view of $\mathcal{B}^*$ is identically distributed to its view when called in the real SWK game. As $\mathcal{B}^*$ gives a correct witness with probability $1 - \nu$, $\mathcal{A}$ rejects with probability ν each of the $\sum_{j=1}^{n} j = \frac{n^2+n}{2}$ times $\mathcal{B}^*$ is called. Assuming it does not halt, then w_{j+1} is a valid witness so the decryption m of C is correct. Hence $\mathcal{A}^*$ is a valid ciphertext extractor for $\mathcal{A}$.

4.4 Kurosawa-Desmedt is Simulatable

Theorem 9. *Assume that X is a simulatable set, that the symmetric encryption scheme is simulatable, that the subset membership problem* **M** *is hard, that* Hash *is δ-smooth and that* **H** *is $(1/|\Pi|)$-universal$_2$. Then the Kurosawa-Desmedt scheme is simulatable.*

The proof is based on the proof that Kurosawa-Desmedt is IND-CCA secure [7].

Since X is a simulatable set, it has a simulation function $Xf : \{0,1\}^{n_X} \to X$. Similarly, the symmetric encryption scheme has a simulation function $Df : \{0,1\}^{n_D} \to \mathcal{C}$. We construct a simulation function for the Kurosawa Desmedt scheme as follows:

$$f : S \times \{0,1\}^{n_X + n_D} \to X \times \mathcal{C}$$
$$(pk, r) \mapsto (Xf(r_1), Df(r_2))$$

where $r = r_1 || r_2$ and r_i is of the appropriate length. Since the ciphertext space is $X \times \mathcal{C}$, this function trivially satisfies the first four conditions of simulatability, using the corresponding conditions for Xf and Df.

We now consider an adversary $\mathcal{A}$ against the fifth simulation property. Let q_e (resp. q_d) be an upper bound for the number of encryption (resp. decryption) queries made by $\mathcal{A}$. We prove that the advantage of $\mathcal{A}$ is negligible using a sequence of games. Let S_i be the event that $\mathcal{A}$ outputs 1 in Game i, and $S_{i,j}$ be the event that $\mathcal{A}$ outputs 1 in Game (i, j).

Game 0: Let Game 0 be the original simulation game, assuming it has access to the real encryption oracle. Written out in full it is as follows:

$(pk, sk) \leftarrow \mathcal{G}(\Lambda)$
$b' \leftarrow \mathcal{A}^{\mathcal{E},\mathcal{D}}(pk)$
Return b'

$\mathcal{A}$ has access to an encryption oracle $\mathcal{E}$ which takes a message m and returns $\mathcal{E}(pk, m)$ and a decryption oracle $\mathcal{D}$ which takes a ciphertext C and returns $\mathcal{D}(sk, C)$. This oracle is restricted so that $\mathcal{A}$ may not request the decryption of any ciphertext which was previously returned by the encryption oracle. We will maintain initially empty lists CLIST which consists of the ciphertexts returned by the encryption oracle, and κLIST which consists of the κ values used by the encryption oracle in their construction.

Game 1: This is a bridging step. We alter the encryption oracle so that it responds to the i^{th} query on input m_i as follows:

$(x_i, w_i) \xleftarrow{R} R$
$\boxed{\pi_i \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x_i)}$
$\kappa_i \leftarrow \mathrm{Hash}(\pi_i)$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x_i, \chi_i)$
CLIST $\leftarrow$ CLIST $\cup \{C_i\}$
κLIST $\leftarrow$ κLIST $\cup \{\kappa_i\}$
Return C_i

This is a bridging step - since Private and Public both compute H_{sk} correctly on L, the encryption oracles behave identically in game 0 and game 1. Thus $\Pr[S_1] = \Pr[S_0]$.

The proof now continues by induction. We make a sequence of modifications to the behaviour of the encryption oracle for the first query, answering all subsequent queries as before, then repeat those modifications for the second query, then the third and so on. The goal is to end up with the encryption and decryption oracles responding as follows:

To respond to the j^{th} encryption query on m_j (for all $1 \leq j \leq q_e$):	To respond to a decryption query on $C = (x, \chi)$:
$x_j \xleftarrow{R} X$ $\kappa_j \xleftarrow{R} \{0,1\}^\ell$ $\chi_j \leftarrow \mathsf{Enc}(\kappa_j, m_j)$ $C_j \leftarrow (x_j, \chi_j)$ $\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$ $\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_i\}$ Return C_j	Parse $\textsc{Clist}$ as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$ Parse $\kappa\textsc{list}$ as $(\kappa_1, \ldots, \kappa_n)$ If $x = x_j$ and $\chi \neq \chi_j$ for some $j < n$ Return $\mathsf{Dec}(\kappa_j, \chi)$ $\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$ $\kappa \leftarrow \mathrm{Hash}(\pi)$ $m \leftarrow \mathsf{Dec}(\kappa, \chi)$ Return m

We define Game $(8, i)$ to be the game where the first i encryption queries are handled as above and all subsequent encryption queries are handled as in game 1. $\mathcal{A}$ respond to a decryption query on $C = (x, \chi)$:

Parse $\textsc{Clist}$ as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$
Parse $\kappa\textsc{list}$ as $(\kappa_1, \ldots, \kappa_n)$
If $x = x_j$ and $\chi \neq \chi_j$ for some $j < \max\{n, i\}$
  Return $\mathsf{Dec}(\kappa_j, \chi)$
$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

The base case, Game $(8, 0)$, is simply Game 1 - all encryption queries are handled as in Game 1, and the decryption oracle also behaves exactly as in game 1, since i is 0.

Game $(2, i)$: This is a bridging step. We alter the encryption oracle so that it responds to the i^{th} query on input m_i as follows:

$\boxed{x_i \xleftarrow{R} X \setminus L}$
$\pi_i \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x_i)$
$\kappa_i \leftarrow \mathrm{Hash}(\pi_i)$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x_i, \chi_i)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_i

In this game, x is chosen at random from $X \setminus L$. By the hardness of the subset membership problem,

$$|\Pr[S_{2,i}] - \Pr[S_{8,i-1}]| \leq \mathbf{Adv}^{\mathrm{SMP}}_{\mathcal{B}} ,$$

where $\mathcal{B}$ is the adversary described below. $\mathcal{B}$ takes an element $x^* \in X$ and runs as follows:

$(pk, sk) \leftarrow \mathcal{G}(\Lambda)$
$b \xleftarrow{\mathrm{R}} \mathcal{A}^{\mathcal{E},\mathcal{D}}(pk)$
Return b

$\mathcal{B}$ responds to an encryption query on input m_j before the i^{th} as the encryption oracle in game $(8, i-1)$ would, i.e:

$x_j \xleftarrow{\mathrm{R}} X$
$\kappa_j \xleftarrow{\mathrm{R}} \{0,1\}^{\ell}$
$\chi_j \leftarrow \mathsf{Enc}(\kappa_j, m_j)$
$C_j \leftarrow (x_j, \chi_j)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_j

To respond to the i^{th} query on input m_i, $\mathcal{B}$ uses the value x^* it was given as input (i.e the challenge value from its SWK challenger):

$\pi_i \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x^*)$
$\kappa_i \leftarrow \mathrm{Hash}(\pi_i)$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x^*, \chi_i)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_i

$\mathcal{B}$ responds to subsequent encryption queries as the encryption oracle in game 1 would, i.e

$(x_i, w_i) \xleftarrow{\mathrm{R}} R$
$\kappa_j \xleftarrow{\mathrm{R}} \{0,1\}^{\ell}$
$\chi_j \leftarrow \mathsf{Enc}(\kappa_j, m_j)$
$C_j \leftarrow (x_j, \chi_j)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_j

To respond to decrypt queries, $\mathcal{B}$ responds as the decryption oracle described in game $(8, i-1)$ would, using the secret key sk, i.e:

Parse CLIST as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$
Parse κLIST as $(\kappa_1, \ldots \kappa_n)$
If $x = x_j$ and $\chi \neq \chi_j$ for some $j < \max\{n, i-1\}$
  Return $\mathsf{Dec}(\kappa_j, \chi)$
$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

It is clear that $\mathcal{B}$ exactly simulates the environment of Game $(8, i-1)$ if $x^* \in L$ or the environment of Game $(2, i)$ if $x^* \notin L$.

By definition,

$$\Pr[1 \leftarrow \mathcal{B}(x^*) | x^* \xleftarrow{\mathrm{R}} L] = \Pr[S_{8,i-1}]$$

and

$$\Pr[1 \leftarrow \mathcal{B}(x^*) | x^* \xleftarrow{\mathrm{R}} X \setminus L] = \Pr[S_{2,i}]\,,$$

so

$$|\Pr[S_{8,i-1}] - \Pr[S_{2,i}]| \leq \mathbf{Adv}^{\mathrm{SMP}}_{\mathcal{B}}$$

as claimed.

Game (3, *i*): We now modify the decryption oracle to treat the i^{th} ciphertext as it would the previous $i-1$, i.e,

Parse CLIST as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$
Parse κLIST as $(\kappa_1, \ldots \kappa_n)$
If $x = x_j$ and $\chi \neq \chi_j$ for some $j < \boxed{\max\{n, i\}}$
  Return $\mathsf{Dec}(\kappa_j, \chi)$
$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

This is a bridging step, since if $x = x_i$, then $\kappa_i = \mathrm{Hash}(\mathsf{Private}_{\mathbf{H}}(sk, x_i))$ by definition, so the result is the same as in Game $(2, i)$. Thus $\Pr[S_{3,i}] = \Pr[S_{2,i}]$.

Game (4, *i*): We modify the decryption oracle again so that if $\mathcal{A}$ submits a ciphertext $C = (x, \chi)$ such that $x \neq x_j$ for all $j \leq i$ and $x \in X \setminus L$, the decryption oracle returns $\perp$, i.e.

Parse CLIST as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$
Parse κLIST as $(\kappa_1, \ldots \kappa_n)$
If $x = x_j$ and $\chi \neq \chi_j$ for some $j < \max\{n, i\}$
  Return $\mathsf{Dec}(\kappa_j, \chi)$
$\boxed{\text{Else If } x \in X \setminus L}$
  $\boxed{\text{Return } \perp}$

$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

Note that the check $x \in X \setminus L$ cannot be implemented in polynomial-time, but this is not necessary as the simulator need not be efficient.

Let $F_{4,i}$ be the event that a ciphertext is rejected by the decryption oracle in Game $(4, i)$ that would not be rejected according to the rules of Game $(3, i)$. We will now show that $F_{4,i}$ is negligible.

$\mathcal{A}$ may gain some information about $\pi_i = H_{sk}(x_i)$ from χ_i, so for simplicity, we will assume that $\mathcal{A}$ has access to π_i. It gains no information about sk from $\mathcal{D}$ queries, since if $x \notin L$, $\mathcal{D}$ rejects, while if $x \in L$, the value $H_{sk}(x)$ is determined by $pk = \alpha(sk)$, which $\mathcal{A}$ has. Thus in the view of $\mathcal{A}$, sk is an element drawn from the uniform distribution on $\{k \in K : H_{sk}(x_i) = \pi_i \wedge pk = \alpha(sk)\}$. We will use the $1/|\Pi|$-universal$_2$ property of $\mathbf{H}$:

For all $x \in X \setminus L$, for all $x^* \in X \setminus \{x\}$, for all $\pi, \pi^* \in \Pi$, for all $pk \in S$ and for sk chosen uniformly at random in K:

$$\Pr[H_{sk}(x) = \pi | sk \xleftarrow{\mathrm{R}} K \wedge H_{sk}(x^*) = \pi^* \wedge \alpha(sk) = pk] \leq 1/|\Pi|$$

Thus for any given query (x, χ) to the decryption oracle such that $x \in X \setminus L$, in the view of $\mathcal{A}$, $\pi = H_{sk}(x)$ is uniformly distributed on Π. Thus by Lemma 2, $\Pr[\mathsf{Dec}(\kappa, \chi) \neq \perp] \leq \epsilon + 2\delta$. So $\Pr[F_{4,i}] \leq q_d(\epsilon + \delta)$. Thus

$$|\Pr[S_{4,i}] - \Pr[S_{3,i}]| \leq q_d(\epsilon + 2\delta)\,.$$

Game $(5, i)$: We modify the encryption oracle so that it responds to the i^{th} query as follows:

$x_i \xleftarrow{\mathrm{R}} X \setminus L$
$\boxed{\pi_i \xleftarrow{\mathrm{R}} \Pi}$
$\kappa_i \leftarrow \mathrm{Hash}(\pi_i)$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x_i, \chi_i)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_i

Let $F_{5,i}$ be the event that x_i is chosen such that $\mathcal{A}$ has already queried the decryption oracle on $C = (x_i, \chi)$. If $F_{5,i}$ occurs, then if $\mathcal{A}$ queries the decryption oracle on C again, it may detect that it is not in the real simulation game, because the decryption oracle will use this new value of κ_i to decrypt C, which may alter the result. $\Pr[F_{5,i}] \leq q_d/|X \setminus L|$ since x_i is chosen at random from $X \setminus L$ and $\mathcal{A}$ makes at most q_d decryption queries.

Assume $F_{5,i}$ does not occur. Then in the view of $\mathcal{A}$, sk is an element drawn from the uniform distribution on $\{k \in K : pk = \alpha(sk)\}$. Since the decryption oracle rejects any queries (x, χ) where $x \in X \setminus L$, it gains no additional information about sk from decryption queries, since the action of H_k is determined by pk on L. It gets no additional information about sk from previous encryption queries either, since the encryption oracle generates x_j and κ_j independently at random. Thus by the $1/|\Pi|$-universal property of $\mathbf{H}$ (which is trivially implied by the $1/|\Pi|$-universal$_2$ property), in $\mathcal{A}$'s view of Game (4,i), π^* is distributed uniformly on Π. Thus the view of $\mathcal{A}$ is unchanged in Game $(5, i)$. So

$$|\Pr[S_{5,i}] - \Pr[S_{4,i}]| \leq \Pr[F_{5,i}] \leq q_d/|X \setminus L|.$$

Game $(6, i)$: We modify the encryption oracle so that it responds to the i^{th} query as follows:

$x_i \xleftarrow{R} X \setminus L$
$\boxed{\kappa_i \xleftarrow{R} \{0,1\}^\ell}$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x_i, \chi_i)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_i

By the δ-smoothness of Hash, we see that if $\kappa_1 \xleftarrow{R} \{0,1\}^\ell$, $\pi \xleftarrow{R} \Pi$ and $\kappa_2 \leftarrow \mathrm{Hash}(\Pi)$, the statistical distance $\Delta[\kappa_1, \kappa_2] \leq \delta$. Thus $\frac{1}{2}|\Pr[1 \leftarrow \mathcal{A}|\kappa \xleftarrow{R} \{0,1\}^\ell] - \Pr[1 \leftarrow \mathcal{A}|\pi \xleftarrow{R} \Pi; \kappa \leftarrow \mathrm{Hash}(\pi)]| \leq \delta$. Thus

$$|\Pr[S_{6,i}] - \Pr[S_{5,i}]| \leq 2\delta\,.$$

Game $(7, i)$: We modify the decryption oracle once more so that it no longer checks if $x \in L$, i.e. it runs as follows:

Parse $\textsc{Clist}$ as $(x_1, \chi_1), \ldots, (x_n, \chi_n)$
Parse $\kappa\textsc{list}$ as $(\kappa_1, \ldots \kappa_n)$
If $x = x_j$ and $\chi \neq \chi_j$ for some $j < \max\{n, i\}$
  Return $\mathsf{Dec}(\kappa_j, \chi)$
$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \mathrm{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

This means it is computable in polynomial-time once more. By the same logic as in Game $(4, i)$,

$$|\Pr[S_{7,i}] - \Pr[S_{6,i}]| \leq q_d(\epsilon + 2\delta)$$

Game $(8, i)$: We modify the encryption oracle so that it responds to the i^{th} query as follows:

$\boxed{x_i \xleftarrow{R} X}$
$\kappa_i \xleftarrow{R} \{0,1\}^\ell$
$\chi_i \leftarrow \mathsf{Enc}(\kappa_i, m_i)$
$C_i \leftarrow (x_i, \chi_i)$
$\textsc{Clist} \leftarrow \textsc{Clist} \cup \{C_j\}$
$\kappa\textsc{list} \leftarrow \kappa\textsc{list} \cup \{\kappa_j\}$
Return C_i

If the subset membership problem is hard, no polynomial-time algorithm can distinguish X from $X \setminus L$.

Proof. Suppose $x_1 \xleftarrow{R} X \setminus L$ and $x_2 \xleftarrow{R} X$, and $x_3 \xleftarrow{R} L$. Then for any distinguishing adversary $\mathcal{C}$, let $|\Pr[1 \leftarrow \mathcal{C}(x_1)] - \Pr[1 \leftarrow \mathcal{C}(x_2)]| = \epsilon$,

$$\begin{aligned}
\Pr[1 \leftarrow \mathcal{C}(x_2)] &= \Pr[1 \leftarrow \mathcal{C}(x_2) | x_2 \in X \setminus L] \Pr[x_2 \in X \setminus L] \\
&\quad + \Pr[1 \leftarrow \mathcal{C}(x_2) | x_2 \in L] \Pr[x_2 \in L] \\
&= \Pr[1 \leftarrow \mathcal{C}(x_2) | x_2 \in X \setminus L] \cdot (1 - \frac{|L|}{|X|}) \\
&\quad + \Pr[1 \leftarrow \mathcal{C}(x_2) | x_2 \in L] \cdot \frac{|L|}{|X|} \\
&= \Pr[1 \leftarrow \mathcal{C}(x_1)] \cdot (1 - \frac{|L|}{|X|}) + \Pr[1 \leftarrow \mathcal{C}(x_3)] \cdot \frac{|L|}{|X|},
\end{aligned}$$

since the distribution of x_2 conditioned on $x_2 \in X \setminus L$ is uniform on $X \setminus L$ and similarly, the distribution of x_2 conditioned on $x_2 \in L$ is uniform on L. So

$$\begin{aligned}
\epsilon &= |\Pr[1 \leftarrow \mathcal{C}(x_1)] - \Pr[1 \leftarrow \mathcal{C}(x_2)]| \\
&= |\Pr[1 \leftarrow \mathcal{C}(x_1)] - (\Pr[1 \leftarrow \mathcal{C}(x_1)] \cdot (1 - \frac{|L|}{|X|}) + \Pr[1 \leftarrow \mathcal{C}(x_3)] \cdot \frac{|L|}{|X|})| \\
&= |\Pr[1 \leftarrow \mathcal{C}(x_1)] \cdot \frac{|L|}{|X|} - \Pr[1 \leftarrow \mathcal{C}(x_3)] \cdot \frac{|L|}{|X|})| \\
&= \frac{|L|}{|X|} \cdot |\Pr[1 \leftarrow \mathcal{C}(x_1)] - \Pr[1 \leftarrow \mathcal{C}(x_3)]|.
\end{aligned}$$

Since $|\Pr[1 \leftarrow \mathcal{C}(x_1)] - \Pr[1 \leftarrow \mathcal{C}(x_3)]|$ is the advantage of $\mathcal{C}$ against the subset membership problem, it is negligible by assumption, hence ϵ is negligible.

In particular, this implies that

$$|\Pr[S_{8,i}] - \Pr[S_{7,i}]| \leq \frac{|L|}{|X|}\mathbf{Adv}^{\text{SMP}}_{\mathcal{C}}$$

This ends the inductive step.

Game 9:

We must return the decryption oracle to its original state. We remove the condition that states that ciphertexts (x, χ) submitted to the decryption oracle with $x = x_j$ are decrypted using κ_j. In other words, a ciphertext (x, χ) submitted to the decryption oracle is decrypted as follows.

$\pi \leftarrow \mathsf{Private}_{\mathbf{H}}(sk, x)$
$\kappa \leftarrow \text{Hash}(\pi)$
$m \leftarrow \mathsf{Dec}(\kappa, \chi)$
Return m

In Game $(9, q_e)$ the value κ_j is randomly chosen; hence, any ciphertext of the form (x_j, χ) will decrypt to $\perp$ with probability $1 - \epsilon$ by the ϵ-rejection secure property of the DEM. In Game 10, the value of π will be randomly distributed on Π by the $(1/|\Pi|)$-universal$_2$ property of $\mathbf{H}$. By lemma 2, the DEM rejects with probability at least $\epsilon + 2\delta$. Therefore, the two games will agree with probability at least $1 - 2(\epsilon + \delta)$.

Hence,

$$|\Pr[S_9] - \Pr[S_{8,q_e}]| \leq 2q_d(\epsilon + \delta)$$

We note that in Dent's proof of plaintext awareness for the Cramer-Shoup scheme, the decryption oracle is not returned to its proper state and the proof is therefore incomplete.

Game 10:

This is a transition based on indistinguishability. We modify encryption oracle again:

$\boxed{x_i \leftarrow Xf(r_1)}$
$\boxed{\chi_i \leftarrow Df(r_2)}$
$C_i \leftarrow (x_i, \chi_i)$
$\text{CLIST} \leftarrow \text{CLIST} \cup \{C_j\}$
$\kappa\text{LIST} \leftarrow \kappa\text{LIST} \cup \{\kappa_j\}$
Return C_i

$|\Pr[S_{10}] - \Pr[S_9]|$ is negligible by definition of a simulatable set.

We now see that $\Pr[S_{11}] = \Pr[1 \leftarrow \mathcal{A}^{\mathcal{FE},\mathcal{FD}}(pk)]$ and $\Pr[S_0] = \Pr[1 \leftarrow \mathcal{A}^{\mathsf{Enc},\mathsf{Dec}}(pk)]$. Thus

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathcal{A}^{\mathcal{E},\mathcal{D}}(1^\ell)] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{FE},\mathcal{D}}(1^\ell)]| \\
&\quad = |\Pr[S_{10}] - \Pr[S_0]| \\
&\quad \leq |\Pr[S_1] - \Pr[S_0]| + q_e(|\Pr[S_{2,i}] - \Pr[S_{8,i-1}]| + \cdots + |\Pr[S_{8,i}] - \Pr[S_{7,i}]|) \\
&\qquad + |\Pr[S_9] - \Pr[S_8]| + |\Pr[S_{10}] - \Pr[S_9]|
\end{aligned}$$

which is negligible as required.

5 Conclusion

We have shown

5.1 Acknowledgements

The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the authors' views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The first author was also funded in part by the EPSRC.

References

1. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.
2. Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2004.
3. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111, 1994.
4. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.
5. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.
6. Alexander W. Dent. The Cramer-Shoup encryption scheme is plaintext aware in the standard model. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 289–307. Springer, 2006.
7. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2004.
8. Victor Shoup. Oaep reconsidered. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259. Springer, 2001.
9. Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2005.
10. Isamu Teranishi and Wakaha Ogata. Relationship between standard model plaintext awareness and message hiding. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 226–240. Springer, 2006.
11. Mark N. Wegman and Larry Carter. New classes and applications of hash functions. In *Foundations Of Computer Science*, pages 175–182. IEEE, 1979.