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 simplified notion of plaintext awareness, which we term PA2I, and show that this is equivalent to the standard definition of PA2 plaintext awareness for encryption schemes that satisfy certain weak security and randomness requirements. We also show that PA2 plaintext awareness is equivalent to PA2+ plaintext awareness under similar security and randomness requirements. This proves a conjecture of Dent that, for suitably random public-key encryption schemes, PA2 plaintext awareness implies PA1+ plaintext awareness.

1 Introduction

Loosely speaking, a public-key encryption scheme is plaintext aware if it is impossible for any reasonable attacker to create a ciphertext without knowing the underlying message. This is an interesting concept, but one that has proven difficult to formalise. The first formal notion of plaintext awareness was introduced by Bellare and Rogaway [3] and later refined by Bellare *et al.* [1]. However, this notion of plaintext awareness could only be achieved in the random oracle model.

Later, Bellare and Palacio [2] introduced a new definition for plaintext awareness. This new notion could be achieved without recourse to the random oracle methodology, yet was consistent with the earlier definitions in the sense that a schemes proven secure under the earlier definition were also secure under the new definition. These new definitions were slightly extended by Dent [4].

In the formal definition, for every ciphertext creator (algorithm) that can output a ciphertext, there should exist a plaintext extractor (algorithm) that can extract the underlying message given all of the inputs of the ciphertext creator (i.e. the explicit inputs and the random coins that the ciphertext creator uses). This is meant to represent the idea that the plaintext extractor can "observe" every action that the ciphertext creator makes when constructing the ciphertext it finally outputs. The plaintext extractor should be able to extract the underlying message of a ciphertext even if the ciphertext creator can query an encryption oracle that provides the ciphertext creator with the encryption of messages that have been drawn from some arbitrary and unknown (polynomial-time) distribution. This is known as PA2 plaintext awareness.

We may also consider a weaker definition in which the ciphertext creator does not have the ability to obtain ciphertexts from the encryption oracle. This is known as PA1 plaintext awareness. Furthermore, the ciphertext creator may also have access to a randomness oracle which returns random bits (PA1+/PA2+ plaintext awareness). This has the effect of making the actions of the ciphertext creator unpredictable in advance. The complexity of these definitions, and the difficulty in achieving the definition using standard computational assumptions, are the two main barriers to the use of plaintext awareness in cryptography.

However, the concept of plaintext awareness has several uses. First, it can be used to show that an encryption scheme is IND-CCA2 secure. It has been proven that an encryption scheme that is PA2 plaintext aware and IND-CPA secure is necessarily IND-CCA2 secure [2]. Second, there are some cryptographic applications which require a scheme to be plaintext aware; for example, the deniable authentication protocol of Di Raimondo, Gennaro and Krawczyk [6]. Lastly, the concept provides an insight into why some public-key encryption schemes are secure, while others are not. We therefore believe that it is an interesting and useful notion to study.

Our Contributions

We attempt to simplify the definition of plaintext awareness. In particular, we introduce a new notion of plaintext awareness in which the ciphertext creator cannot obtain the encryption of messages drawn from an arbitrary and unknown distribution, but only the encryption of messages drawn from a simple, fixed distribution. This distribution is defined by the plaintext creator $\mathcal{P}_I$ which takes two messages as input and chooses one of those messages at random. We term this new notion of plaintext awareness PA2I as this is precisely the distribution of messages that one considers when proving IND security.

We show that for encryption schemes meeting certain weak security and randomness requirements (IND-CPA security, OW-CPA security and γ-uniformity) the notions of PA2, PA2I and PA2+ plaintext awareness are equivalent. This equivalence proves a conjecture of Dent [4] that a suitably random PA2 plaintext aware encryption scheme is necessarily PA1+ plaintext aware. As a by-product of these theorems, we also show that an encryption scheme that is IND-CPA and PA2 plaintext aware must satisfy the stronger property that an adversary cannot distinguish between encryptions of messages of different lengths, a property not required by the standard definition of indistinguishability. In particular, this implies that the scheme has a finite message space. Finally, we show that PA2I plaintext awareness is not equivalent to PA2 plaintext awareness if the encryption scheme is only OW-CPA secure and γ-uniform.

2 Definitions

2.1 Notation

We will use the following notation in this paper. If S is a set, then $x \overset{R}{\leftarrow} S$ means x is sampled uniformly at random from the set S. If S is a distribution, then

$x \xleftarrow{\text{R}} S$ means that x is sampled according to the distribution. For a deterministic 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 $x \leftarrow \mathcal{A}^{\mathcal{O}}(y, z; R)$ to mean the output of $\mathcal{A}$ when run on inputs y and z with oracle access to $\mathcal{O}$ and using the random coins R. If we do not specify R then we implicitly assume that the coins are selected uniformly at random from $\{0,1\}^{\infty}$. This is denoted $x \xleftarrow{\text{R}} \mathcal{A}^{\mathcal{O}}(y, z)$. We let $R[\mathcal{A}]$ denote the coins of an algorithm $\mathcal{A}$.

2.2 Public-Key Encryption Schemes

An encryption scheme is a triple $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ of probabilistic polynomial-time algorithms. The algorithm $\mathcal{G}(1^{\lambda})$ outputs a key pair (pk, sk). The public key pk implicitly defines a message space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$. The encryption algorithm takes as input a public key pk and a message $m \in \mathcal{M}$, and outputs a ciphertext $C \in \mathcal{C}$. The decryption algorithm takes as input a private key sk and a ciphertext $C \in \mathcal{C}$, and outputs either a message $m \in \mathcal{M}$ or the unique 'reject' symbol $\perp$. We require that if $(pk, sk) \xleftarrow{\text{R}} \mathcal{G}(1^{\lambda})$, then for all $m \in \mathcal{M}$

$$\Pr[\mathcal{D}(sk, \mathcal{E}(pk, m)) = m] = 1 \,.$$

where the probability is taken over the random coins of the encryption algorithm.

We will refer to a public-key encryption scheme as having either a finite or infinite message space. A public-key encryption scheme Π has an infinite message space if $\mathcal{M}$ is an infinite set for all values of the security parameter λ. Π has a finite message space if $\mathcal{M}$ is a finite set for all values of the security parameter λ. For simplicity, we will assume that all public-key encryption schemes either have the infinite message space $\mathcal{M} = \{0,1\}^*$ (as with most hybrid encryption schemes) or the finite message space $\mathcal{M} = \{0,1\}^{\ell(\lambda)}$. We will assume that all encryption schemes run in time that is polynomially bounded in the size of their inputs (i.e. λ and $|m|$).

Note that if $\ell(\lambda)$ is polynomially bounded then we may equivalently define a finite message space as $\mathcal{M} = \{0,1\}^{<\ell}$, i.e. the set of all bit strings of length less than ℓ, as there is a trivial polynomial-time map from $\{0,1\}^{<\ell}$ into $\{0,1\}^{\ell}$.

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:

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

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

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ATK}} = |\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]| \, .$$

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^*.

Definition 1 (IND-ATK). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is IND-ATK secure if for any probabilistic, polynomial-time IND-ATK adversary $\mathcal{A}$, the advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ATK}}$ is negligible as a function of λ.*

Frequently, where it will not cause undue confusion, we will suppress the state information STATE and simply assume that all necessary information is passed from $\mathcal{A}_1$ to $\mathcal{A}_2$.

2.4 One-Wayness

We also require a notion of one-wayness (OW-CPA) for an encryption scheme with an infinite message space. For simplicity we assume that $\mathcal{M} = \{0, 1\}^*$. One-wayness is assessed via the following game:

$(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$
$m \overset{R}{\leftarrow} \{0, 1\}^\lambda$
$C^* \overset{R}{\leftarrow} \mathcal{E}(pk, m)$
$m' \overset{R}{\leftarrow} \mathcal{A}(pk, C^*)$

We define the attacker $\mathcal{A}$'s success probability to be $\Pr[m' = m]$.

Definition 2 (OW-CPA). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is OW-CPA secure if for any probabilistic polynomial-time OW-CPA adversary $\mathcal{A}$, the success probability of $\mathcal{A}$ is negligible as a function of λ.*

2.5 Plaintext Awareness

The formal definition of plaintext awareness in the standard model was proposed by Bellare and Palacio [2]. A scheme is plaintext aware if for every probabilistic polynomial-time algorithm (ciphertext creator) $\mathcal{A}$ there exists a probabilistic polynomial-time algorithm (plaintext extractor) $\mathcal{A}^*$ which can simulate a decryption oracle for $\mathcal{A}$ when given the random coins that $\mathcal{A}$ uses (in the sense that the output of $\mathcal{A}$ when interacting with $\mathcal{A}^*$ is computationally indistinguishable from the output of $\mathcal{A}$ when interacting with a real decryption oracle). In order to model the attacker's ability to obtain ciphertexts for which it does not know the underlying decryption, the ciphertext creator is equipped with an oracle that will return the encryption of a randomly chosen message $m \overset{R}{\leftarrow} \mathcal{P}(s)$ where $\mathcal{P}$ is an arbitrary probabilistic polynomial-time algorithm (plaintext creator) and s is supplied by the ciphertext creator $\mathcal{A}$. Note that both $\mathcal{P}$ and $\mathcal{A}^*$ are considered to be stateful algorithms.

Formally, we consider two games. In both cases, the ciphertext creator $\mathcal{A}$ is given a public key pk from a correctly generated public-key pair $(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$ and outputs a bitsting x. In both cases, the ciphertext creator has access to an "encryption oracle" that will, on input s, generate a message $m \overset{R}{\leftarrow} \mathcal{P}(s)$, compute $C \overset{R}{\leftarrow} \mathcal{E}(pk, m)$, add C to a list of returned ciphertexts Clist and return C to the ciphertext creator. The games are distinguished by the "decryption oracle" to which $\mathcal{A}$ has access. In the Real game, $\mathcal{A}$ can query a decryption oracle on any ciphertext $C \notin \textsc{Clist}$ and the oracle will return $\mathcal{D}(sk, C)$. In the Fake game, $\mathcal{A}$ can query a decryption oracle on any ciphertext $C \notin \textsc{Clist}$ and the oracle will execute $\mathcal{A}^*(pk, C, R[\mathcal{A}], \textsc{Clist})$ and return the result. We stress again that $\mathcal{A}^*$ and $\mathcal{P}$ are stateful algorithms. We can summarise these two games as follows:

Real game:

$$(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$$
$$x_{\text{Real}} \overset{R}{\leftarrow} \mathcal{A}^{\mathcal{D}(sk,\,\cdot\,),\mathcal{E}(pk,\mathcal{P}(\,\cdot\,))}(pk)$$

Fake game:

$$(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$$
$$x_{\text{Fake}} \overset{R}{\leftarrow} \mathcal{A}^{\mathcal{A}^*(pk,\,\cdot\,,R[\mathcal{A}],\textsc{Clist}),\mathcal{E}(pk,\mathcal{P}(\,\cdot\,))}(pk)$$

Definition 3 (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 $\mathcal{P}$ and polynomial-time distinguishing algorithms D, the advantage*

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

is negligible as a function of the security parameter (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 (PA1). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA1 plaintext aware if it is PA2 plaintext aware for all ciphertext creators $\mathcal{A}$ that do not make any queries to the encryption oracle. In other words, Π is PA1 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 distinguishing algorithms D, the advantage*

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

is negligible as a function of the security parameter.

Dent [4] extended these definitions to allow the ciphertext creator $\mathcal{A}$ to take actions that are unpredictable to the plaintext extractor $\mathcal{A}^*$ in advance by allowing the ciphertext creator $\mathcal{A}$ to repeatedly query a "randomness oracle" which returns a single random bit.

Definition 5 (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.* $\mathcal{A}^*(pk, C, R[\mathcal{A}], \textsc{Rlist}, \textsc{Clist})$.

Note that any such PA+ definition implies the corresponding PA definition, since an adversary may simply not use the randomness oracle.

Bellare and Palacio proved that [2] any scheme that was PA2 plaintext aware and IND-CPA secure was IND-CCA2 secure. The proof of this fact makes use of a particular plaintext creator $\mathcal{P}_I$ which takes as input two messages (m_0, m_1) and outputs a randomly chosen message m_b. We call this the IND plaintext creator and define a scheme to be PA2I plaintext aware if it is PA2 plaintext aware for the IND plaintext creator.

Definition 6 (PA2I). *A public key encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2I 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 distinguishing algorithms D, the advantage*

$$\mathbf{Adv}^{PA2I}_{\mathcal{A},\mathcal{A}^*,D} = \mathbf{Adv}^{PA2}_{\mathcal{A},\mathcal{A}^*,\mathcal{P}_I,D}$$

is negligible as a function of the security parameter.

The paper of Bellare and Palacio [2] actually proves that a scheme which is PA2I plaintext aware and IND-CPA secure is IND-CCA2 secure. We note that a theorem of Teranishi and Ogata [8] shows that any scheme which is one-way and PA2 plaintext aware is IND-CCA2 secure. We stress that the proof of Teranishi and Ogata requires the use of the arbitrary plaintext creator $\mathcal{P}$ provided by the full definition of PA2 plaintext awareness.

3 Theoretical Results About Plaintext Awareness

3.1 Connection Between PA2I And PA2

One of the more complex aspects of plaintext awareness is the fact that the encryption oracle returns an encryption of a message that has been chosen from some arbitrary distribution defined by $\mathcal{P}$. The order of the quantifiers in the definition of PA2 plaintext awareness means that neither the ciphertext creator $\mathcal{A}$, nor the plaintext extractor $\mathcal{A}^*$, know the distribution from which messages are chosen, although the ciphertext creator does have the ability to affect this distribution via its input s to the encryption oracle. In this section, we show that for IND-CPA encryption schemes it is sufficient to consider the fixed plaintext creator $\mathcal{P}_I$. We note that PA2 plaintext awareness trivially implies PA2I plaintext awareness, so we will concentrate on proving the converse theorem.

Theorem 1. *If an encryption scheme with the finite message space $\mathcal{M} = \{0,1\}^{\ell(\lambda)}$ is IND-CPA secure and PA2I plaintext aware, and $\ell(\lambda)$ is polynomially bounded in λ, 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 but for each value of λ we have that $\ell(\lambda)$ is finite.

Proof. Consider an arbitrary plaintext creator $\mathcal{P}$. We prove that the output of $\mathcal{A}$ interacting with $\mathcal{P}$ is computationally indistinguishable from the output of $\mathcal{A}$ interacting with $\mathcal{P}_I$ and therefore, if there exists a plaintext extractor $\mathcal{A}^*$ for the ciphertext creator $\mathcal{A}$ in the PA2I model, then $\mathcal{A}^*$ is also a plaintext extractor for the ciphertext creator $\mathcal{A}$ in the PA2 model. We prove this through a sequence of four games. Let x_i be the output of $\mathcal{A}$ in Game i. 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 $\mathcal{P}$. In other words, the encryption oracle computes messages $m \overset{R}{\leftarrow} \mathcal{P}(s)$ and returns $C \overset{R}{\leftarrow} \mathcal{E}(pk, m)$. The decryption oracle returns $\mathcal{A}^*(pk, C, R[\mathcal{A}], \text{CLIST})$.

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. Since $\ell(\lambda)$ is polynomially bounded, this action can be computed in polynomial time. The oracle then encrypts this message, then returns the ciphertext to $\mathcal{A}$ and adds it to CLIST.

If $|\Pr[S_1] - \Pr[S_0]|$ is non-negligible, then we can construct an adversary $\mathcal{B}$ that breaks the IND-CPA security of the scheme. We use a simple hybrid argument. Suppose $\mathcal{A}$ makes at most q_e queries to the encryption oracle. $\mathcal{B}_1$ takes as input the public key pk and runs $\mathcal{A}$ and $\mathcal{A}^*$ exactly as described in the Game 0. $\mathcal{B}$ responds to the first $q_e - 1$ encryption oracle queries as in Game 0 (i.e. by computing a message $m \overset{R}{\leftarrow} \mathcal{P}(s)$ and returning $C \overset{R}{\leftarrow} \mathcal{E}(pk, m)$). For the q_e-th query to the encryption oracle, $\mathcal{B}_1$ generates both $m_0 \overset{R}{\leftarrow} \mathcal{P}(s)$ and $m_1 \overset{R}{\leftarrow} \mathcal{P}_I(s)$ and outputs (m_0, m_1) as the messages on which it wishes to be challenged.

The challenger will pick one of these messages and encrypt it, the result will be returned to $\mathcal{B}_2$. $\mathcal{B}_2$ handles any decryption oracle queries by $\mathcal{A}$ in the same way as before (i.e. by using $\mathcal{A}^*$). Eventually $\mathcal{A}$ terminates and outputs a bitstring x. $\mathcal{B}_2$ terminates by outputting the bit $D(x)$.

Since Π is IND-CPA, $\mathcal{B}$'s advantage is bounded by $\mathbf{Adv}_{\mathcal{B}}^{\text{IND-CPA}}$. It is clear that if the challenger chose to encrypt message m_0, then $\mathcal{A}$ was playing Game 0. It also clear that if the challenger chose to encrypt message m_1 then $\mathcal{A}$ was playing a hybrid game in which the first $q_e - 1$ queries were answered as in Game 0 and the last query was answered as in Game 1. Hence, the probability that the ciphertext creator $\mathcal{A}$ outputs a bitstring x such that $D(x) = 1$ can only change by at most $\mathbf{Adv}_{\mathcal{B}}^{\text{IND-CPA}}$ if the final encryption is computed using $\mathcal{P}_I$ rather than $\mathcal{P}$.

We now repeat this "trick" q_e times, until all the encryption oracle queries are handled as in Game 1. Hence,

$$|Pr[S_1] - Pr[S_0]| \leq q_e \mathbf{Adv}_{\mathcal{B}}^{\text{IND-CPA}} .$$

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

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

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 as in Game 1, except that this time we construct an IND-CCA2 adversary $\mathcal{B}$, which uses its own decryption oracle to answer decryption queries. We may assume that Π is IND-CCA2 secure as it is both IND-CPA secure and PA2I plaintext aware. Hence, after q_e rounds, we have that

$$|\Pr[S_3] - \Pr[S_2]| \leq q_e \mathbf{Adv}^{\text{IND-CCA}}_{C}$$

Note that Game 3 is identical to the Real game with plaintext creator $\mathcal{P}$. We can therefore conclude that

$$\begin{aligned}\mathbf{Adv}^{\text{PA2}}_{\mathcal{A},\mathcal{A}^*,\mathcal{P},D} &= |Pr[S_0] - Pr[S_3]| \\ &\leq q_e \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{B}} + \mathbf{Adv}^{\text{PA2I}}_{\mathcal{A},\mathcal{A}^*,D} + q_e \mathbf{Adv}^{\text{IND-CCA}}_{\mathcal{B}}\end{aligned}$$

Since the scheme is PA2I and IND-CPA, we see that

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathcal{B}} \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{C}} + q_d \mathbf{Adv}^{\text{PA2I}}_{\mathcal{F},\mathcal{F}^*,D'}$$

for some probabilistic polynomial time algorithms $\mathcal{C}, \mathcal{F}, \mathcal{F}^*$ and D'. Thus

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

which is negligible as required. □

Corollary 1. *If an encryption scheme Π is IND-CPA secure and PA2I+ plaintext aware then it is PA2+ plaintext aware.*

Proof. The proof of this theorem mirrors the proof of Theorem 1. □

The fact that we may be substitute an arbitrary plaintext creator $\mathcal{P}$ with the specific plaintext creator $\mathcal{P}_I$ will be crucial in proving the relationship between PA2 and PA2+ in Section 3.3.

For schemes that have already been shown to be IND-CCA2 secure, but about which their plaintext awareness may be in doubt, we can prove a stronger result. Let $\mathcal{P}_m$ be the plaintext creator that constantly outputs the message $m \in \mathcal{M}$.

Corollary 2. *If an encryption scheme Π is IND-CCA2 secure and PA2 (resp. PA2+) plaintext aware with respect to the specific plaintext creator $\mathcal{P}_m$, then it is PA2 (resp. PA2+) plaintext aware.*

Proof. The proof of this theorem mirrors the proof of Theorem 1 except we explicitly use the fact that Π is IND-CCA2 secure in the third game hop, rather than deriving the fact that Π is IND-CCA2 secure from the fact that it is IND-CPA secure and PA2I plaintext aware. □

This corollary may have some applications in situations where public key encryption schemes are known to be IND-CCA2 secure, but need to be shown to be PA2 plaintext aware in order that they might be used in some specific protocol, e.g. the deniable authentication protocol of Di Raimondo, Gennaro and Krawczyk [6].

3.2 PA2 and One-Wayness Implies a Finite Message Space

In the previous section, we introduced an extra condition into our proof – we required the encryption scheme to have a finite message space. This may seem like an unreasonable restriction. Far from being unreasonable, particularly when one considers hybrid encryption schemes; however, we will show in this section that a finite message space is necessary in order for a one-way scheme to achieve PA2 plaintext awareness. Hence, we can conclude that many hybrid encryption schemes, are unable to achieve this level of security, at least if we define the message space to be $\{0,1\}^*$, the set of all bitstrings. Our proof will not preclude the possibility that a scheme is PA2I plaintext aware, OW-CPA secure and has an infinite message space.

Theorem 2. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. If Π is PA2 and has an infinite message space, then it is not OW-CPA.*

In order to prove this theorem, we use the proof technique of Teranishi and Ogata [8]. The technique involves using a specific plaintext creator $\mathcal{P}$ to leak the value of a ciphertext C^* to the ciphertext creator $\mathcal{A}$ bit-by-bit in such a way that C^* does not appear on CLIST. The plaintext creator can then query the decryption oracle on C^* to obtain the underlying message (the validity of which it can check using one further query to the plaintext creator). Now, since this system allows the ciphertext creator to decrypt an arbitrary ciphertext by interacting with only the polynomial-time plaintext extractor, the encryption scheme cannot be one-way. Our proof differs from Teranishi and Ogata in that we will leak the value of the challenge ciphertext C^* by outputting short ciphertexts if a bit of C^* is zero and long ciphertexts if a bit of C^* is one. We can produce ciphertexts which are recognisably short or long due to the infinite size of the message space.

Proof. We will prove that if $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is PA2 and has an infinite message space then Π is not OW-CPA secure. For simplicity, we assume $\mathcal{M} = \{0,1\}^*$.

Note that the length of any ciphertext must be 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 $l_0 = f(\lambda,\lambda) + \lambda + 1$, $l_1 = f(\lambda, l_0) + \lambda + 1$, 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 and returns

$$\mathsf{Decode}(C) = \begin{cases} 0 & \text{if } f(\lambda,\lambda) < |C| \le f(\lambda, l_0) \\ 1 & \text{if } f(\lambda, l_0) < |C| \le f(\lambda, l_1) \\ 2 & \text{if } f(\lambda, l_1) < |C| \le f(\lambda, l_2) \\ \perp & \text{otherwise} \end{cases}$$

If $C \xleftarrow{R} \mathcal{E}(pk, \mathsf{Encode}(0))$, then we would like $\mathsf{Decode}(C) = 0$. However, since we only know that $|C| \le f(\lambda, l_0)$, it is possible that $|C| \le f(\lambda,\lambda)$ and so the decode algorithm will fail. But, since there exists only $2^{f(\lambda,\lambda)+1} - 1$ ciphertexts of length at most $f(\lambda, \lambda)$ and $2^{l_0} - 1$ messages of length l_0, the probability that a randomly chosen message will encrypt to give a ciphertext of length less than or equal to $f(\lambda, \lambda)$ is bounded by $2^{-\lambda}$. Similarly, the probability that $\mathsf{Decode}(C) \neq i$ when $C \xleftarrow{\text{R}} \mathcal{E}(pk, \mathsf{Encode}(i))$ for $i \in \{1, 2\}$ is bounded by $2^{-\lambda}$.

Next we construct a ciphertext creator $\mathcal{A}$ and a specific plaintext creator $\mathcal{P}$. The plaintext creator $\mathcal{P}$ works in a series of phases:

1. The first time the plaintext creator is initialised it picks a random message $m^* \xleftarrow{\text{R}} \{0,1\}^\lambda$ and computes $C^* \xleftarrow{\text{R}} \mathcal{E}(pk, m)$.
2. For the i-th query, where $1 \leq i \leq |C^*|$, the plaintext creator returns $\mathsf{Encode}(b_i)$, where b_i is the i-th bit of C^*. Hence, the ciphertext creator will receive $\mathcal{E}(pk, \mathsf{Encode}(b_i)$. This leaks the value of the ciphertext C^* to the ciphertext creator.
3. For the next query the plaintext creator returns $\mathsf{Encode}(2)$. This signifies the end of the ciphertext.
4. For the next query the plaintext creator uses the input s provided by the ciphertext creator. If $s = m^*$ then the ciphertext creator returns $\mathsf{Encode}(1)$; otherwise it returns $\mathsf{Encode}(0)$. This is a validity check.
5. For all subsequent queries the plaintext creator outputs 0.

The ciphertext creator $\mathcal{A}$ works as follows:

1. The ciphertext creator queries the plaintext creator repeatedly, each time receiving a ciphertext C and computing the bit $b \leftarrow \mathsf{Decode}(C)$. If $b \in \{0, 1\}$ then the ciphertext creator stores this bit and repeats the query. If $b = 2$ then the ciphertext creator continues to the next phase.
2. The ciphertext creator reconstructs the ciphertext C^* from the bits recovered in the first phase.
3. The ciphertext creator submits the ciphertext C^* to the decryption oracle and receives a message m.
4. Next, the ciphertext creator submits m to the encryption oracle and receives back a ciphertext C.
5. The ciphertext creator outputs the bit $\mathsf{Decode}(C)$

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. We note that if the decode algorithm always returned the correctly encoded bit, then $C^* \notin$ CLIST as every ciphertext C that the encryption oracle returns is of size greater than $f(\lambda, \lambda)$. Furthermore, if the decode algorithm always returned the correctly encoded bit, the $\mathcal{A}$ will always return 1 in the REAL game. Hence,

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

Since, Π is PA2 plaintext aware, there exists a plaintext extractor $\mathcal{A}^*$ for the ciphertext creator $\mathcal{A}$ with the property that

$$Pr[S_{fake}] \geq 1 - (|C^*| + 2) \cdot 2^{-\lambda} - \mathbf{Adv}^{\text{PA2}}_{\mathcal{A}, \mathcal{A}^*, \mathcal{P}, D}$$

where D is the trivial distinguishing algorithm that outputs the single bit which it takes as input. Due to the validity check, this means that $\mathcal{A}^*$ must return the correct decryption of C^* with probability $Pr[S_{fake}]$.

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

1. $\mathcal{B}$ receives a ciphertext C^* and sets n to be $|C^*|$.
2. $\mathcal{B}$ generates a simulation of $\textsc{Clist} \leftarrow \{C_0, C_1, \ldots, C_{n+1}\}$ in which $C_i \overset{R}{\leftarrow} \mathcal{E}(pk, \mathsf{Encode}(b_i))$, for $1 \leq i \leq n$ and where b_i is the i-th bit of C^*, and $C_{n+1} \overset{R}{\leftarrow} \mathcal{E}(pk, \mathsf{Encode}(2))$.
3. $\mathcal{B}$ generates a suitably large random tape $R[\mathcal{A}]$. The useable tape length can be polynomially bounded by the runtime of $\mathcal{A}^*$; hence, the construction of such a tape is polynomial time.
4. $\mathcal{B}$ computes $m \overset{R}{\leftarrow} \mathcal{A}^*(pk, C^*, R[\mathcal{A}], \textsc{Clist})$ and returns m.

Since $\mathcal{B}$ exactly simulates the environment in which $\mathcal{A}^*$ runs, $\mathcal{B}$ correctly decrypts C^* with probability $Pr[S_{fake}] \geq 1 - (|C^*| + 2) \cdot 2^{-\lambda} - \mathbf{Adv}^{\text{PA2}}_{\mathcal{A},\mathcal{A}^*,\mathcal{P},D}$ which is non-negligible as required. □

This proof actually shows that any PA2 plaintext-aware encryption scheme which a message space $M = \{0,1\}^{<\ell(\lambda)}$ cannot be OW-CPA if $\ell(\lambda)$ grows faster than any polynomial. This is because we only require that the message space be able to cope with messages up to length $l_2(\lambda)$ for the proof to work.

We may also conclude that any public-key encryption scheme Π which is IND-CPA secure, PA2I plaintext aware and has an infinite message space cannot be PA2 plaintext aware (as in such a case IND-CPA security implies OW-CPA security and this contradicts the previous theorem). Hence, the condition that the message space be finite in Theorem 1 is necessary.

3.3 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 randomised encryption scheme, since an adversary may use randomness inherent in a ciphertext generated by the encryption oracle to simulate a randomness oracle. This in turn implies that a suitably random PA2 encryption scheme is PA1+, thus giving a formal proof to the conjecture of Dent [4].

The proof essentially involves constructing a randomness oracle by taking ciphertexts created by a γ-uniform encryption algorithm and hashing them onto a single bit using a randomly chosen universal$_2$ hash function. The resulting distribution on $\{0,1\}$ is only a small statistical distance from the uniform distribution on $\{0,1\}$ and the result follows from the Leftover Hash Lemma [5]. One subtlety of the proof is that we will require the ciphertext creator $\mathcal{A}^*$ that we construct to know the functionality of the plaintext creator $\mathcal{P}$. Hence, we actually prove that a suitably random PA2I plaintext aware encryption scheme is PA2I+, and appeal to Theorem 1 to finish the proof.

Definition 7 (γ-Uniformity). *An encryption scheme is γ-uniform if for all public keys pk, messages m and ciphertexts C, $\Pr[\mathcal{E}(pk,m) = C] \leq \gamma$, where the probability is taken over the choice of random coins used by the encryption 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, $\Pr[H_k(x) = H_k(y) | k \overset{R}{\leftarrow} K] \leq 1/|B|$.*

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 [9].

Definition 9 (Statistical Distance). *Let x and y be random variables taking values on a finite set S. We define the statistical distance between x and y as*

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

Note that if $\mathcal{A}$ is a predicate on the set S, then the following inequalities holds:

$$\varDelta[x,y] \geq |\Pr[\mathcal{A}(x)] - \Pr[\mathcal{A}(y)]| \tag{1}$$

We give the version of Leftover Hash Lemma given in Theorem 6.21 of [7].

Lemma 1 (Leftover Hash Lemma). *Let $\mathbf{H}$ be a family of universal$_2$ hash functions from A to B where B is of size β. Let V denote any distribution on A which is independent of the choice of k. Let $\hat{U}$ and $\hat{V}$ denote the distributions given by*

$$\hat{U} = \{(k,y) : k \overset{R}{\leftarrow} K,\, y \overset{R}{\leftarrow} B\} \qquad \hat{V} = \{(k,y) : k \overset{R}{\leftarrow} K,\, x \overset{R}{\leftarrow} V,\, y \leftarrow H_k(x)\}$$

and let

$$\kappa = \sum_{a\in A} Pr[V = a]^2 \,.$$

Then $\varDelta[\hat{U},\hat{V}] \leq \sqrt{\beta\kappa}/2$.

This allows us to prove the following lemma.

Lemma 2. *Let $\varPi$ be a γ-uniform encryption scheme, then, for any fixed message $m \in \mathcal{M}$ and public key pk, we have*

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

where the probability is taken over the choice of $k \overset{R}{\leftarrow} \{0,1\}^n$ and the random coins used by the encryption algorithm.

Proof. Let V be the distribution of $C \xleftarrow{R} \mathcal{E}(pk, m)$. By the γ-uniformity of Π we have

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

So

$$\kappa(V) \quad \leq \sum_{v \in \{0,1\}^*} \Pr[C = v]\gamma \quad = \quad \gamma \sum_{v \in \{0,1\}^*} \Pr[C = v] \quad = \quad \gamma$$

and so by the Leftover Hash Lemma we have

$$\Delta[(k, H_k(C)), (k, y)] \leq \sqrt{2\gamma}/2\,,$$

where $y \xleftarrow{R} \{0,1\}$. However,

$$\Delta[(k, H_k(C)), (k, y)] \geq |Pr[H_k(C) = 1] - 1/2|$$

which gives the required result. □

Theorem 3. *Suppose a public key encryption scheme Π is γ-uniform (for a negligible value of γ) and PA2I plaintext aware. Then it is PA2I+ plaintext aware.*

Proof. Let $\mathbf{H}$ be as above. Let $\mathcal{A}$ be a PA2I+ ciphertext creator that makes at most q_r queries to the randomness oracle. We construct a PA2I 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 $\mathcal{A}(pk; R[\mathcal{A}])$. $\mathcal{B}$ 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 encryption oracle on the 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 PA2I ciphertext creator, there exists a plaintext extractor $\mathcal{B}^*$. We use $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$ for $\mathcal{A}$.

Recall that $\mathcal{A}^*$ takes input $(pk, C, R[\mathcal{A}], \textsc{Rlist}, \textsc{Clist})$. We will assume that when $\mathcal{A}^*$ is first initialised it chooses hash keys $(k_1, \ldots, k_{q_r}) \xleftarrow{R} (\{0,1\}^n)^{q_r}$ and stores these keys. If $\mathcal{A}^*$ is queried with a ciphertext C, then it runs as follows:

1. If the randomness oracle has been queried since $\mathcal{A}^*$ was last executed, i.e. Rlist has grown, then for each new bit b_i that has been returned $\mathcal{A}^*$ generates a ciphertext C_i by running $\mathcal{E}(pk, \mathcal{P}_I(0))$ repeatedly until it finds C_i such that $H_{k_i}(C_i) = b_i$, then adds C_i to Clist in the appropriate place. We note that, by Lemma 2, the probability that $Pr[H_{k_i}(C) \neq b_i] \leq \frac{1}{2} + \sqrt{\gamma/2}$. We limit $\mathcal{A}^*$ to running λ trials; hence, $\mathcal{A}^*$ will run in polynomial time, but fail with the negligible probability $(\frac{1}{2} + \sqrt{\gamma/2})^\lambda$.
2. $\mathcal{A}^*$ then computes $m \xleftarrow{R} \mathcal{B}^*(pk, C, R, \textsc{Clist})$ where $R = k_1|| \ldots ||k_{q_r}||R[\mathcal{A}]$.

We now show that $\mathcal{A}^*$ is a valid plaintext extractor for $\mathcal{A}$, i.e. the output $x \xleftarrow{R} \mathcal{A}^{\mathcal{O}}(pk)$ is computationally indistinguishable in the Real and Fake games.

Fix a distinguishing algorithm D, let x_i be the output of $\mathcal{A}$ in Game i and let S_i be the event that $D(x_i) = 1$.

Game 0: Let Game 0 be the Real game for $\mathcal{A}$. In other words, the encryption oracle takes as input s, computes $m \overset{\text{R}}{\leftarrow} \mathcal{P}_I(s)$ and returns $C \overset{\text{R}}{\leftarrow} \mathcal{E}(pk, m)$. The decryption oracle returns $\mathcal{D}(sk, C)$.

Game 1: We modify the randomness oracle so that on the i^{th} query it computes $C_i \overset{\text{R}}{\leftarrow} \mathcal{E}(pk, \mathcal{P}_I(0))$ and sets $b_i \leftarrow H_{k_i}(C_i)$, where $1 \leq i \leq q_r$, rather than simply returning a random bit. In order to prove that $|\Pr[S_0] - \Pr[S_1]|$ is negligible, we use a hybrid argument. Suppose we consider changing the response of the first query to the randomness oracle from the random bit b to the bit $b' \overset{\text{R}}{\leftarrow} H_{k_1}(\mathcal{E}(pk, \mathcal{P}_I(0)))$ and let S^* be the event that $D(x) = 1$ in this new game. By Lemma 2 and Equation 1 we have that

$$|\Pr[S_0] - \Pr[S^*]| \quad \leq \quad \Delta[(k_1, b), (k_1, b')] \quad \leq \quad \sqrt{\gamma/2}$$

We may repeat this argument for all q_r randomness oracle queries to obtain

$$|\Pr[S_0] - \Pr[S_1]| \leq q_r \sqrt{\gamma/2}$$

Game 2: We modify the randomness oracle so that it adds each ciphertext $C_i \overset{\text{R}}{\leftarrow} \mathcal{P}_I(0)$ it generates to Clist. Since the ciphertext creator $\mathcal{A}$ does not have access to Clist and the ciphertext creator $\mathcal{A}$ has access to a real decryption oracle, the view of $\mathcal{A}$ is identical in the two games unless it submits one of these ciphertexts to the decryption oracle. The probability that a specific ciphertext involved in a decryption oracle query matches a specific ciphertext created by the randomness oracle is bounded by γ due to the γ-uniformity property. Since $\mathcal{A}$ makes at most q_r randomness oracle queries such ciphertexts and at most q_d decryption queries, we have

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

Game 3: We modify the decryption oracle so that it uses the plaintext extractor $\mathcal{A}^*$ to answer decryption oracle queries. Game 3 exactly simulates the environment of $\mathcal{B}^*$ providing that the $\mathcal{B}^*$ finds a suitable ciphertext C_i for each random bit b_i on Rlist, so if D is an arbitrary distinguishing algorithm for $\mathcal{B}$,

$$|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\text{PA2I}}_{\mathcal{B},\mathcal{B}^*,\mathcal{P},D} + q_r(\frac{1}{2} + \sqrt{\gamma/2})^{\lambda}$$

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

$$\begin{aligned}\mathbf{Adv}^{\text{PA2I+}}_{\mathcal{A},\mathcal{A}^*,\mathcal{P},D} &= |\Pr[S_3] - \ \Pr[S_0]| \\ &\leq \mathbf{Adv}^{\text{PA2I}}_{\mathcal{B},\mathcal{B}^*,\mathcal{P},D} + q_r q_e \gamma + q_r\sqrt{\gamma/2} + q_r(\frac{1}{2} + \sqrt{\gamma/2})^{\lambda}\,.\end{aligned}$$

which is negligible as required. □

Corollary 3. *Suppose a public key encryption scheme Π is PA2 plaintext aware, OW-CPA secure, and γ-uniform. Then Π is PA2+ plaintext aware.*

Proof. Since Π is PA2 plaintext aware and OW-CPA secure, we have that it is PA2I plaintext aware, IND-CPA secure and that it has a finite message space $\mathcal{M} = \{0,1\}^{\ell(\lambda)}$ where $\ell(\lambda)$ is polynomially bounded (Theorem 2). Since Π is PA2I plaintext aware and γ-uniform, we have that it is PA2I+ plaintext aware (Theorem 3). Since Π has a finite message space and is both PA2I+ plaintext aware and IND-CPA secure, we have that it is PA2+ plaintext aware (Corollary 1). □

3.4 PA2I+ And OW-CPA Do Not Guarantee IND-CPA Security

We have shown that for IND-CPA encryption schemes, the notions of PA2I plaintext awareness and PA2 plaintext awareness are equivalent. It might be hoped that this equivalence also holds for schemes with fewer security guarantees – in particular, it might be hoped that one can find an analogue of the Teranishi and Ogata theorem [8] which would prove that a scheme which was PA2I plaintext aware and OW-CPA secure was IND-CCA2 secure.

In this section we give evidence that this is not the case by proving that there exist schemes that are PA2I+ plaintext aware and OW-CPA secure, but which are not IND-CPA secure. Alternatively, by Theorem 3, we have that there exists a scheme which is PA2I plaintext aware, OW-CPA secure and γ-uniform, but not IND-CPA secure. We leave the question of showing that there exists schemes that are PA2I plaintext aware and OW-CPA secure, but not IND-CPA secure, as an open problem.

Theorem 4. *Suppose there exists a public key encryption encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ 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 assume that the message space $\mathcal{M}$ for Π is such that it is easy to find messages m_0 and m_1 which differ in the final bit and let $F(m)$ denote the final bit of message m. We now describe a new encryption scheme $\Pi' = (\mathcal{G}, \mathcal{E}', \mathcal{D}')$ as follows:

```
$\mathcal{E}'(pk, m)$:
    $C' \overset{R}{\leftarrow} \mathcal{E}(pk, m)$
    $b \leftarrow F(m)$
    $C \leftarrow (C', b)$
    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 two messages (m_0, m_1) that differ in the final bit.

We now show that Π' is PA2I+. Let $\mathcal{A}$ be a PA2I+ ciphertext creator against Π'. We construct a PA2I+ ciphertext creator $\mathcal{B}$ against Π. $\mathcal{B}$ runs $\mathcal{A}(pk; R[\mathcal{B}])$ and handles queries as follows:

- If $\mathcal{A}$ makes an encryption oracle query on (m_0, m_1), $\mathcal{B}$ queries its own encryption 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 $C = (C', F(m_0))$ to $\mathcal{A}$. If not, $\mathcal{B}$ queries its randomness oracle to get a bit b', and returns $C = (C', b')$.
- If $\mathcal{A}$ makes a decryption query on $C = (C', b')$, $\mathcal{B}$ checks whether $(C', b' \oplus 1)$ is on Clist. If so, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ 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 the ciphertext creator $\mathcal{B}$. We use $\mathcal{B}^*$ to construct a plaintext extractor $\mathcal{A}^*$ for the ciphertext creator $\mathcal{A}$. $\mathcal{A}^*$ takes input $(pk, C, R[\mathcal{A}], \text{Rlist}, \text{Clist})$ and runs as follows:

1. When it is first initialised, $\mathcal{A}^*$ creates two empty lists Rlist' and Clist' which will be used to simulate the inputs to the plaintext extractor $\mathcal{B}^*$.
2. $\mathcal{A}^*$ checks to see if the encryption oracle or decryption oracle has been used since it was last activated. It does this by executing $\mathcal{A}$ on all the appropriate inputs (using pk, $R[\mathcal{A}]$ and the values on Clist and Rlist).
   - For each new bit b' returned by the randomness oracle, $\mathcal{A}^*$ appends b' to Rlist'.
   - For each new ciphertext (C', b') returned by the encryption oracle, $\mathcal{A}^*$ examines the two messages (m_0, m_1) that $\mathcal{A}$ submitted to the encryption oracle (which $\mathcal{A}^*$ knows because it has executed $\mathcal{A}$). If $F(m_0) = F(m_1)$, then $\mathcal{A}^*$ appends C' to Clist'. If $F(m_0) \neq F(m_1)$, then $\mathcal{A}^*$ appends b' to Rlist' and C' to Clist'.
3. If $C \in \text{Clist}'$ then $\mathcal{A}^*$ returns $\perp$.
4. Otherwise, $\mathcal{A}^*$ computes $m \xleftarrow{R} \mathcal{B}^*(pk, C, R[\mathcal{A}], \text{Rlist}', \text{Clist}')$.
5. If $F(m) = b'$ then $\mathcal{A}^*$ returns m; otherwise $\mathcal{A}^*$ returns $\perp$.

We must now show that $\mathcal{A}^*$ is a valid plaintext extractor for $\mathcal{A}$. We do this by showing that $\mathcal{A}$ and $\mathcal{A}^*$ almost perfectly simulates the output of $\mathcal{B}$ and $\mathcal{B}^*$. Fix a distinguishing algorithm D, let x_i be the output of $\mathcal{A}$ in Game i and let S_i be the event that $D(x_i) = 1$ in Game i.

Game 0: Let Game 0 be the Real game for $\mathcal{A}$. In other words, the encryption oracle takes as input two messages (m_0, m_1), chooses a bit $b \xleftarrow{R} \{0, 1\}$ and returns $(C', b') \xleftarrow{R} \mathcal{E}'(pk, m_b)$. The decryption oracle returns $\mathcal{D}'(sk, C)$.

Game 1: We let Game 1 be identical to Game 0 except that for each ciphertext (C', b') returned by the encryption oracle, the bit b' is chosen in the same way that $\mathcal{B}$ does – i.e. if $F(m_0) = F(m_1)$ then the oracle chooses $b' = F(m_0)$, 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

$$|\Pr[S_1] - \Pr[S_0]| \leq q_e \mathbf{Adv}^{\text{IND-CCA2}}_{\mathcal{B}'}$$

for some IND-CCA2 adversary $\mathcal{B}'$ against $\mathit{\Pi}$, since if the outputs of $\mathcal{A}$ are distinguishable in these two games, we can construct an adversary which distinguishes ciphertexts. Note that we may assume $\mathit{\Pi}$ is IND-CCA2 secure as it is IND-CPA secure and PA2I+ plaintext aware.

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 $\mathcal{B}$. Thus by the PA2I+ property of $\mathit{\Pi}$,

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

Game 3: Let Game 3 be as Game 2, except with the original behaviour of the encryption oracle restored, i.e. the final bit of the ciphertext is the final bit of the message. Hence,

$$|\Pr[S_3] - \Pr[S_2]| \leq q_e \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{B}'}$$

for some IND-CPA adversary $\mathcal{B}'$ for the same reasoning as in Game 1.

However, Game 3 is identical to the Fake game for $\mathcal{A}$. Hence,

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

which is negligible as required. □

4 Conclusion

In this paper we have discussed the relationship between several notions of computational plaintext awareness, most notably the relationship between PA2 and the newly introduced notion of PA2I. The relationships between PA2I and PA2 are summarised in the diagram below:

$$\begin{tikzcd}
 & \text{IND-CPA} & \text{OW-CPA} \\
\text{PA2} & \bullet \arrow[r, shift left, "1"] \arrow[d, shift right] & \bullet \arrow[l, shift left, "2"] \arrow[d] \\
\text{PA2I} & \bullet \arrow[r, shift left, "1"] \arrow[u, shift right, "3"'] & \bullet \arrow[l, shift left, "/"]
\end{tikzcd}$$

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, since in this case any scheme which is IND-CPA is also OW-CPA. The arrow numbered 2 follows from the result of Teranishi and Ogata [8]. The arrow numbered 3 is a result of Theorem 1 and the separation is a result of Theorem 4 (under the added assumption that the encryption scheme is γ-uniform). Note that the diagram also demonstrates that there exist schemes that are OW-CPA, γ-uniform and PA2I, but not PA2. We believe that in almost all practical cases, the PA2I notion of plaintext awareness suffices.

We also explored some of the properties of encryption schemes that are PA2 plaintext aware, γ-uniform, OW-CPA secure and IND-CPA secure. We demonstrated that these schemes must have a finite message space and that they are necessarily PA2+. This latter result proves the conjecture of Dent [4].

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. 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.
5. Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24. ACM, 1989.
6. Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. Deniable authentication and key exchange. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 400–409. ACM, 2006.
7. Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2005.
8. 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.

9. Mark N. Wegman and Larry Carter. New classes and applications of hash functions. In *Foundations Of Computer Science*, pages 175–182. IEEE, 1979.