On Ciphertext Undetectability

Peter Gaži and Martin Stanek *

Dept. of Computer Science, Comenius University
Mlynská dolina, 842 48 Bratislava, Slovakia
{gazi,stanek}@dcs.fmph.uniba.sk

Abstract. We propose a novel security notion for public-key encryption schemes – ciphertext undetectability. Informally, an encryption scheme has the property of ciphertext undetectability, if the attacker is unable to distinguish between valid and invalid ciphertexts. We compare this notion with the established ones, such as indistinguishability of ciphertexts and plaintext awareness. We analyze the possibilities of constructing schemes with the property of ciphertext undetectability. Moreover, we prove that the Damgård ElGamal, the Cramer-Shoup scheme and its lite variant achieve ciphertext undetectability under standard assumptions.

Keywords: ciphertext undetectability, public-key encryption

1 Introduction

In the framework of provable security, there are various established security notions for public-key encryption schemes. The security notions evolved, from simple one-wayness into plaintext awareness, to reflect gradually increasing requirements we impose on encryption schemes. Usually, a security notion is obtained by combining an adversarial goal, such as indistinguishability or non-malleability, and an attack model, such as chosen plaintext attack or (adaptive) chosen ciphertext attack. Another approach is to require that the attacker is unable to produce a valid ciphertext without the knowledge of the corresponding plaintext (leading to various forms of plaintext awareness). For relations among standard security notions, see [3].

We propose a novel adversarial goal – ciphertext undetectability. Informally, an encryption scheme has the property of ciphertext undetectability, if the polynomial time probabilistic attacker is unable to distinguish between valid and invalid ciphertexts. The motivation behind this security notion is three-fold:

– In contrast to plaintext awareness, distinguishing valid ciphertexts might be, for the attacker, easier than actually producing them.

* This work was supported in part by the grant VEGA 1/3106/06.

– Similar concepts were already studied and proved to be useful. Hall, Goldberg, and Schneier [9] used an oracle for distinguishing valid ciphertexts (validity checking oracle) in the attack model (reaction attack). Dent [8] defined a much stronger notion of encryption simulatability in order to prove the plaintext awareness of the Cramer-Shoup system.
– Ciphertext undetectability can be an interesting property per-se. Besides purely theoretical view, it can be viewed as a "steganographic" property of an encryption scheme – inability to distinguish actual ciphertexts from some random elements.

Since ciphertext undetectability (CUD for short) describes an adversarial goal, we can pair it with the well-known attack models, namely chosen plaintext attack (CPA), (non-adaptive) chosen ciphertext attack (CCA1), and adaptive chosen ciphertext attack (CCA2).

Contribution. We give a formal definition of CUD, and analyze the properties of the resulting security notions. Namely:

– We prove a separation of CUD-CPA/CCA1/CCA2 classes. More importantly, we show that CUD is an independent security notion for public-key encryption schemes by proving separation of CUD-*atk* and security notions IND-CPA/CCA1/CCA2 and PA. Since CUD-*atk* does not imply even IND-CPA, see Theorem 4, it can hardly provide any form of secrecy alone. However, CUD can be viewed as an additional, non-trivial property for public-key encryption schemes. Since PA does not imply CUD-CPA, see Theorem 5, the CUD property is not guaranteed even for PA secure schemes.
– We prove a separation of CUD-*atk* and usual public-key steganographic security notion of ciphertext pseudorandomness.
– We discuss and analyze various approaches to construct schemes with the CUD property, namely doubling technique, and padding. These techniques allow to turn ElGamal, Goldwasser-Micali, or other IND-CPA secure schemes into CUD-CPA secure variants.
– We prove the CUD-CPA security of Cramer-Shoup lite scheme, and DEG (Damgård ElGamal) scheme under the DDH assumption. In addition, we prove CUD-CPA security of Cramer-Shoup scheme under the DDH assumption, and CUD-CCA1 security of Cramer-Shoup lite scheme and DEG scheme under the DDH and DHK1 assumptions.

Related work. Plaintext awareness (PA, see [3]) is the strongest security notion for public-key encryption schemes. Informally, any attacker in the PA secure scheme is unable to produce a valid ciphertext without the knowledge of the corresponding plaintext. The attacker's task can be easier when his goal is not to produce a valid ciphertext, but recognize valid ciphertexts. The idea of distinguishing valid and invalid ciphertext was used in the reaction attacks [9] or validity-checking attacks [11], where the attacker can use this ability (modeled as an oracle) to attack the scheme. On the contrary, our approach treats

the validity-checking as a goal of the attacker. We can look at the attacker as a "censor", who knows your public key and tries to block all your ciphertexts (but not the invalid ciphertexts).

Steganographic properties of public-key encryption schemes were studied in [1, 2, 10]. The idea of ciphertext pseudorandomness (or "real-or-ideal" property) is that the attacker chooses a message and then is unable to distinguish some encryption of this message from a random string of the same length. This notion differs from our notion of CUD. In fact, they are incomparable, as we will prove later.

Dent defined similar, but much stronger notion of simulatable encryption [8], in order to prove the plaintext awareness of the Cramer-Shoup scheme in the standard model. Dent proved that each simulatable encryption scheme is IND-CCA2 secure. CUD can be viewed as a simpler and weaker version of simulatable encryption.

The paper is structured as follows. Section 2 contains notions and definitions used in the paper. In addition, we define the security notion of ciphertext undetectability. In Section 3 we prove the separation of the CUD-*atk* classes. The relations of CUD-*atk* to established security notions of IND-*atk*, PA and ciphertext pseudorandomness are analyzed in Section 4. We discuss the constructions of CUD-*atk* secure schemes and CUD-properties of DEG scheme and Cramer-Shoup schemes in Section 5.

2 Preliminaries

Throughout the paper, we use the well-known notions, typical for the provable security framework. We call a function $f\colon \mathbb{N} \to \mathbb{R}$ negligible, if for every constant $c \geq 0$ there exists an integer k_c such that $f(k) \leq k^{-c}$ for all $k \geq k_c$. By a shorthand $x \stackrel{R}{\leftarrow} M$ we denote assigning to x a random value from the set M. We call two families of sets $\{X_k\}_{k\in\mathbb{N}}$ and $\{Y_k\}_{k\in\mathbb{N}}$ computationally indistinguishable and denote this by a shorthand $\{X_k\}_{k\in\mathbb{N}} \approx_c \{Y_k\}_{k\in\mathbb{N}}$ (or $X \approx_c Y$ if the parameter k is obvious) if, for any probabilistic polynomial algorithm A, there exists a negligible function ε such that, for any k:

$$\Pr[x \stackrel{R}{\leftarrow} X_k; b \leftarrow A(x) : b = 1] - \Pr[y \stackrel{R}{\leftarrow} Y_k; b \leftarrow A(y) : b = 1] \leq \varepsilon(k).$$

We denote an asymmetric encryption scheme by the usual triple $S = (G, E, D)$, where G is the randomized key generation algorithm, E is the randomized encryption algorithm and D is the deterministic decryption algorithm. The set of all plaintexts is denoted by $\mathcal{P}$, so $\mathcal{P} = Dom(E)$. The symbol $\mathcal{C}$ denotes the set $Dom(D)$ while it is required to hold $E(\mathcal{P}) \subseteq \mathcal{C}$ and $\forall p \in \mathcal{P};\ D(E(p)) = p$ for all random coins of E. We denote by $\mathcal{C}_v$ the set of all valid ciphertexts, i.e. $\mathcal{C}_v = \{c \in \mathcal{C} \mid D(c) \neq \perp\}$.

We define ciphertext undetectability using an experiment. Let A be a probabilistic polynomial adversary A running in two phases: *ask* and *guess*. In the *ask* phase, A can do precomputations, possibly with an access to the decryption

oracle (depending on the attack model atk). As a result, it outputs some state information s for its second phase. Then a fair coin is tossed and depending on the result, either a valid or invalid ciphertext from the superset $\overline{\mathcal{C}}$ of $\mathcal{C}_v$ is chosen as a challenge. This challenge is presented to the adversary and its task is to guess the result of the coin toss. The experiment is formalized as follows:

$\texttt{Experiment}\ \ \mathbf{Expt}^{\mathrm{cud-atk-b}}_{S,\overline{\mathcal{C}},A}(k)$
$(pk, sk) \leftarrow G(1^k)$;
$s \leftarrow A^{D_1(.)}(\mathrm{ask},pk)$;
if $b = 1$ **then** $y \xleftarrow{R} \mathcal{C}_v$
else $y \xleftarrow{R} \overline{\mathcal{C}} \setminus \mathcal{C}_v$;
$b' \leftarrow A^{D_2(.)}(\mathrm{guess},y,s)$;
return b';

attack model	$D_1(.)$	$D_2(.)$
CPA	$\perp$	$\perp$
CCA1	$D(.)$	$\perp$
CCA2	$D(.)$	$D(.)$

where $b \in \{0, 1\}$, $atk \in \{CPA, CCA1, CCA2\}$ and the oracles D_1 and D_2 are instantiated – reflecting the attack model – according to the table. As usually, if $atk = CCA2$ we do not allow A to try to decrypt the challenge y. We define the advantage of an adversary to be

$$\mathbf{Adv}^{\mathrm{cud-atk}}_{S,\overline{\mathcal{C}},A}(k) = \Pr[\mathbf{Expt}^{\mathrm{cud-atk-1}}_{S,\overline{\mathcal{C}},A}(k) = 1] - \Pr[\mathbf{Expt}^{\mathrm{cud-atk-0}}_{S,\overline{\mathcal{C}},A}(k) = 1]$$

and the insecurity of the scheme (relative to a given set $\overline{\mathcal{C}}$) to be

$$\mathbf{InSec}^{\mathrm{cud-atk}}_{S,\overline{\mathcal{C}}}(k) = \max_A \mathbf{Adv}^{\mathrm{cud-atk}}_{S,\overline{\mathcal{C}},A}(k)$$

where A ranges through all polynomial time probabilistic adversaries. These notions are defined to correspond with analogous notions defined for indistinguishability in [3]. We also refer to [3] for the definition of indistinguishability and plaintext awareness in the random oracle model.

Definition 1. *An asymmetric encryption scheme S is CUD-atk secure ($atk \in \{CPA, CCA1, CCA2\}$) if there exists a set $\overline{\mathcal{C}}$ such that:*

1. $\mathcal{C}_v \subsetneq \overline{\mathcal{C}}$
2. *there exists a deterministic polynomial algorithm, which accepts exactly the set $\overline{\mathcal{C}}$*
3. *the function $\boldsymbol{InSec}^{\mathrm{cud-atk}}_{S,\overline{\mathcal{C}}}(.)$ is negligible.*

Intuitively, an encryption scheme is CUD secure, if its valid ciphertexts cannot be distinguished from other members of some larger set, possibly even with a (limited) access to the decryption oracle. In the specific case of CUD-CPA, this corresponds to the well-known notion of computational indistinguishability of sets $\mathcal{C}_v$ and $\overline{\mathcal{C}} \setminus \mathcal{C}_v$ with respect to the security parameter of the scheme.

For most of the real encryption schemes, there exists an implicitly given set $\overline{\mathcal{C}}$ satisfying the properties 1 and 2. Obviously, if $\mathcal{C}_v$ can be easily recognized, there exists no set $\overline{\mathcal{C}}$ with the required properties, therefore the scheme is not CUD-atk secure for any attack scenario atk.

Example 1. The set $\mathcal{C}_v$ is easily recognizable for textbook versions of RSA, ElGamal, or Goldwasser-Micali scheme, hence none of these schemes if CUD-*atk* secure.

3 Separation of the CUD-*atk* Classes

It trivially holds CUD-CCA2 $\Rightarrow$ CUD-CCA1 $\Rightarrow$ CUD-CPA. In this section, we shall prove the separations for the other direction.

Theorem 1 (CUD-CPA $\not\Rightarrow$ CUD-CCA1). *If there exists a CUD-CPA secure scheme, then there exists a CUD-CPA secure scheme which is not CUD-CCA1 secure.*

Proof. Let $S = (G, E, D)$ be a CUD-CPA secure scheme. Let us consider the scheme $S' = (G', E', D')$:

$\texttt{Algorithm } G'(1^k)$
$u \xleftarrow{R} \{0,1\}^k$
$(pk, sk) \leftarrow G(1^k)$
return $((pk, u), sk)$

$\texttt{Algorithm } E'_{pk,u}(m)$
return $0 \| E_{pk}(m)$

$\texttt{Algorithm } D'_{sk}(c')$
parse c' **as** $b \| c$
if $b = 0$ **return** $D(c)$
else if $c = u$ **return** sk
else return $\perp$

S' is obviously not CUD-CCA1 secure. Any adversary who asks the provided oracle to decrypt the ciphertext $1\|u$ obtains the private key sk and can decide validity of arbitrary messages.

However, S' remains CUD-CPA secure. Let $\mathcal{C}_v$ be the set of valid ciphertexts in S, let $\overline{\mathcal{C}}$ be the corresponding superset according to the definition of CUD-CPA for S. Then the set of valid ciphertexts in S' is obviously $\mathcal{C}'_v = 0\|\mathcal{C}_v \cup \{1\|u\}$. We shall show that $\overline{\mathcal{C}}' = 0\|\overline{\mathcal{C}} \cup \{1\|u\}$ can be used as the required set proving that S' is CUD-CPA secure. It is easy to see that $\mathcal{C}'_v$ and $\overline{\mathcal{C}}'$ satisfy properties 1 and 2 of Definition 1. To prove the last property, suppose there exists an adversary A' capable of distinguishing valid ciphertexts of S' from the other members of $\overline{\mathcal{C}}'$. Then the following adversary A would be capable of doing the same for S and the set $\overline{\mathcal{C}}$.

$\texttt{Algorithm } A(\text{ask}, pk)$
$u \xleftarrow{R} \{0,1\}^k$
$s \leftarrow A'(\text{ask}, (pk, u))$
return s

$\texttt{Algorithm } A(\text{guess}, y, s)$
$b' \leftarrow A'((\text{guess}, 0\|y, s)$
return b'

The value u is random and therefore useless without access to the decryption oracle. A answers correctly whenever A' does, so their advantage only differs according to the behavior of A' on the input $1\|u$, but this minor difference is clearly negligible. □

Theorem 2 (CUD-CCA1 $\not\Rightarrow$ CUD-CCA2). *If there exists a CUD-CCA1 secure scheme, then there exists a CUD-CCA1 secure scheme which is not CUD-CCA2 secure.*

Proof. Let $S = (G, E, D)$ be a CUD-CCA1 secure scheme. Let us consider $S' = (G, E', D')$:

```
Algorithm E'_pk(m)
  return 0‖E_pk(m)

Algorithm D'_sk(c')
  parse c' as b‖c
  return D_sk(c)
```

S' is obviously not CUD-CCA2 secure – an adversary can decide validity of any challenge $0\|c$ by querying the decryption oracle for $1\|c$.

However, S' remains CUD-CCA1 secure. Let $\mathcal{C}_v$ denote the set of valid ciphertexts of S and let $\overline{\mathcal{C}}$ be the corresponding superset guaranteed by the CUD-CCA1 definition for S. Then the set of valid ciphertexts of S' is $\{0,1\}\|\mathcal{C}_v$, let us show that $\{0,1\}\|\overline{\mathcal{C}}$ can be used as the corresponding superset. Suppose there exists an adversary A' that can distinguish valid ciphertexts of S' from the other members of $\{0,1\}\|\overline{\mathcal{C}}$. Then the following adversary A can do the same for S and $\overline{\mathcal{C}}$.

```
Algorithm A^{D(.)}(ask, pk)
  s ← A'^{D'(.)}(ask, pk)
  return s

Algorithm A(guess, s, c)
  d ←R {0,1}
  b' ← A'(guess, s, d‖c)
  return b'
```

Obviously A can use the oracle $D(.)$ to perfectly simulate the oracle $D'(.)$ for A' during its first phase. A answers correctly whenever A' does, therefore the advantages of both adversaries are the same. □

4 Relations to established notions

In this section, we shall inspect the relationship between ciphertext undetectability and some well-known security notions: indistinguishability, various forms of plaintext awareness and ciphertext pseudornadomness. Obviously, we cannot expect ciphertext undetectability to imply secrecy. It turns out that neither does indistinguishability imply CUD. However, the notion of plaintext awareness in the standard model defined in [4] plays a similar role when combined with CUD, as it does when combined with indistinguishability.

4.1 CUD and Indistinguishability

Let us first inspect the relationship between ciphertext undetectability and indistinguishability.

Theorem 3 (IND-*atk* $\not\Rightarrow$ CUD-CPA). *For any $atk \in \{CPA, CCA1, CCA2\}$, if there exists an IND-atk secure scheme then there exists an IND-atk secure scheme which is not CUD-CPA secure.*

Proof. Let $atk \in$ {CPA,CCA1,CCA2} be an attack model, let $S = (G, E, D)$ be an IND-*atk* secure scheme. If S is CUD-CPA secure, let $\overline{\mathcal{C}}$ be the corresponding

superset of $\mathcal{C}_v$. Let us consider a new scheme $S' = (G', E, D')$, where G' and D' are modified in the following way

```
Algorithm G'(1^k)
  (pk, sk) ← G(1^k)
  p_0 ←R P
  return ((pk, p_0), (sk, p_0))

Algorithm D'_(sk,p_0)(c')
  if c' ∈ C̄ then
    if D_sk(c') = ⊥ then return p_0
    else return D_sk(c')
  else return ⊥
```

and the encryption algorithm works as before, using the public key pk. Intuitively, D' returns an arbitrary, valid, publicly known plaintext p_0 whenever the original decryption procedure D would fail on a member of $\overline{\mathcal{C}}$. This can be done, since the membership in $\overline{\mathcal{C}}$ can be easily decided. Note that this modification does not violate the basic condition of correct decryption, posed on all encryption schemes.

According to our argument following Definition 1, the new scheme S' is not CUD-CPA secure, since its set of valid ciphertexts is now $\overline{\mathcal{C}}$, which is easily recognizable.

However, S' remains IND-*atk* secure. It would be easy to transform any successful IND-*atk* attacker on S' into an IND-*atk* attacker on S, because the D' oracle can be perfectly simulated, having acces to the D oracle. Therefore, by our assumption on S, S' is also IND-*atk* secure. □

Theorem 4 (CUD-*atk* $\not\Rightarrow$ IND-CPA). *For any* $atk \in \{CPA, CCA1, CCA2\}$, *if there exists a CUD-*atk *secure encryption scheme, then there exists a CUD-*atk *secure scheme which is not IND-CPA secure.*

Proof. Let $atk \in$ {CPA,CCA1,CCA2} be an attack model, let $S = (G, E, D)$ be a CUD-*atk* secure scheme. Consider a new scheme $S' = (G', E', D)$:

```
Algorithm G'(1^k)
  (pk, sk) ← G
  m+ ←R P
  c+ ← E_pk(m+)
  return ((pk, m+, c+), sk)

Algorithm E'_(pk,m+,c+)(m)
  if m = m+ return c+
  else return E_pk(m)
```

The new scheme S' is not IND-CPA secure, since its behavior on a publicly known plaintext m^+ is deterministic.

However, S' is CUD-*atk* secure. To see this, it suffices to observe that both the sets of valid ciphertexts and the decryption oracles of S and S' are the same. Therefore, any successful CUD-*atk* adversary for S' would also be a successful CUD-*atk* adversary for S. □

4.2 CUD and Plaintext Awareness

Let us examine the relationship of ciphertext undetectability and various forms of plaintext awareness. First, let us consider the concept of plaintext awareness in

the random oracle model, as defined in [3], which we shall denote PA. Although PA is a very strong security notion, it does not imply ciphertext undetectability, as the next theorem shows.

Theorem 5 (PA $\not\Rightarrow$ CUD-CPA). *If there exists a PA encryption scheme, then there exists a PA scheme which is not CUD-CPA secure.*

Proof. We can use the same construction as in the proof of Theorem 3. Let S be a PA encryption scheme, let S' be the scheme obtained from S by this construction. As before, S' is no longer CUD-CPA secure. Since S was IND-CPA secure (as a part of PA), the proof of Theorem 3 implies that so is also S'. Moreover, any successful plaintext extractor A for S can be easily transformed into a plaintext extractor A' for S' that works in the same way as A does, but instead of returning $\perp$ on a member of $\overline{\mathcal{C}}$, it returns p_0. Since S is plaintext aware, A exists, and therefore also A' can be constructed. This proves the plaintext awareness of S'. □

Now let us consider the forms of plaintext awareness in the standard model, defined in [4]. The following theorem shows the intuitive fact that PA1 accompanied with CUD-CPA plays the same role as it does for indistinguishability, i.e. it prevents any adversary from exploiting the pre-challenge queries by making them useless.

Theorem 6 (CUD-CPA + PA1 $\Rightarrow$ CUD-CCA1). *If S is a PA1 and CUD-CPA secure encryption scheme, then S is CUD-CCA1 secure.*

Proof. The intuitive argument goes as follows: if there exists a successful plaintext extractor capable of answering the pre-challenge decryption queries without knowing the private key (based only on the random tape of the adversary), then this plaintext extractor can be used as a subroutine of the adversary to answer the queries. Therefore, a successful CUD-CCA1 adversary does not need to use its pre-challenge decryption oracle and can be used as a CUD-CPA adversary, with only a negligible difference in its advantage.

The proof is similar to the original proof of the claim IND-CPA + PA1 $\Rightarrow$ IND-CCA1 in [4] and therefore is omitted. □

4.3 CUD and Ciphertext Pseudorandomness

The notion of ciphertext pseudorandomness is usually discussed in steganographic public-key encryption schemes [1, 2, 10]. First, let us define the ciphertext pseudorandomness using the standard game-based approach:

$\texttt{Experiment}\ \mathbf{Expt}^{\mathrm{cpr-atk-b}}_{S,A}(k)$
$(pk, sk) \leftarrow G(1^k)$;
$(m, s) \leftarrow A^{D_1(\cdot)}(\mathrm{ask}, pk)$;
if $b = 1$ **then** $y \xleftarrow{R} \{0,1\}^{n(k)}$
else $y \xleftarrow{R} E(m)$;
$b' \leftarrow A^{D_2(\cdot)}(\mathrm{guess}, y, s)$;
return b';

where $b \in \{0,1\}$, $n(k)$ denotes the length of the ciphertext (it is assumed that encryption of a plaintext of some length always results in a ciphertext of fixed length), $atk \in \{CPA, CCA1, CCA2\}$ and the oracles D_1 and D_2 are instantiated according to the attack model. As usually, if $atk = CCA2$ we do not allow A to try to decrypt the challenge y. We define the advantage of an adversary to be

$$\mathbf{Adv}^{\mathrm{cpr-atk}}_{S,A}(k) = \Pr[\mathbf{Expt}^{\mathrm{cpr-atk-1}}_{S,A}(k) = 1] - \Pr[\mathbf{Expt}^{\mathrm{cpr-atk-0}}_{S,A}(k) = 1]$$

and the insecurity of the scheme to be

$$\mathbf{InSec}^{\mathrm{cpr-atk}}_{S}(k) = \max_{A} \mathbf{Adv}^{\mathrm{cpr-atk}}_{S,A}(k)$$

where A ranges through all polynomial time probabilistic adversaries.

We say that S is CPR-*atk* secure if $\mathbf{InSec}^{\mathrm{cpr-atk}}_{S}(\cdot)$ is negligible. Thus, an attacker is unable to distinguish a ciphertext (corresponding to the chosen plaintext) from a completely random string of the same length.

Although the notion of ciphertext pseudorandomness ("real-or-ideal") seems similar to CUD ("valid-or-invalid"), we show that these notions are incomparable.

Theorem 7 (CUD-*atk* $\not\Rightarrow$ CPR-CPA). *For any $atk \in \{CPA, CCA1, CCA2\}$, if there exists a CUD-*atk *secure encryption scheme, then there exists a CUD-*atk *secure scheme which is not CPR-CPA secure.*

Proof. We use the construction from the proof of Theorem 4. The scheme S' is not CPR-CPA secure, since its behavior on a publicly known plaintext m^+ is deterministic. Thus, the attacker can choose m^+ as a plaintext and distinguish its encryption from a random string with overwhelming probability. □

Theorem 8 (CPR-*atk* $\not\Rightarrow$ CUD-CPA). *For any $atk \in \{CPA, CCA1, CCA2\}$, if there exists an CPR-atk secure scheme then there exists an CPR-atk secure scheme which is not CUD-CPA secure.*

Proof. We use the construction from the proof of Theorem 3. The scheme S' remains CPR-*atk* secure (if the original scheme S was CPR-*atk* secure), since the decryption oracle D' can be simulated perfectly, having access to the D oracle. Hence any successful CPR-*atk* attacker on S' can be transformed into a successful CPR-*atk* attacker on S. □

5 Constructions and CUD-*atk* Secure Schemes

5.1 Doubling technique

The doubling technique allows us to turn an IND-CPA secure scheme into an IND-CPA secure scheme with the CUD-CPA property. The idea of the doubling technique is to form the ciphertext from two independent encryptions of the plaintext. In order to preserve the IND-CPA security of the resulting scheme, we need the property of ciphertext self-reducibility.

Definition 2. *An asymmetric encryption scheme $S = (G, E, D)$ has the property of ciphertext self-reducibility, if there exists publicly known probabilistic polynomial time algorithm R, such that $R(c)$ returns a uniformly distributed value from the set $\{c' \mid D(c') = D(c)\}$, for any $c \in \mathcal{C}_v$.*

The property of encryption self-reducibility allows anyone to transform a ciphertext into another ciphertext encrypting the same plaintext.

Example 2. The following schemes have the property of self-reducibility (some other schemes, such as Cramer-Shoup lite and other have this property as well):

ElGamal scheme. Let g be a generator of a group $\mathbb{G}$ of order q, the private key is $x \xleftarrow{R} \mathbb{Z}_q$, the public key is $y = g^x$. The encryption of $m \in \mathbb{G}$ is the pair $(u, v) = (g^r, y^r m)$, where $r \xleftarrow{R} \mathbb{G}$. The ciphertext is easily self-reducible: $R(u, v) = (u \cdot g^l, v \cdot y^l)$, where $l \xleftarrow{R} \mathbb{Z}_q$.

Goldwasser-Micali scheme. Let n be a product of two large primes. Let y be a pseudo-square modulo n, i.e. a quadratic non-residue modulo n with Jacobi symbol 1. The values n and y form a public key, the secret key is the factorization of n. The encryption of $m \in \{0, 1\}$ is $y^m r^2 \bmod n$, where $r \xleftarrow{R} \mathbb{Z}_n^*$. The ciphertext is self-reducible: $R(c) = c \cdot l^2 \bmod n$, where $l \xleftarrow{R} \mathbb{Z}_n^*$.

The doubling technique transforms a scheme $S = (G, E, D)$ into $S' = (G, E', D')$ as follows:

```
Algorithm E'_pk(m)                Algorithm D'_sk(c1 || c2)
  return E_pk(m) || E_pk(m)         if D_sk(c1) = D_sk(c2) then return D_sk(c1)
                                    return ⊥
```

In the following theorems, we show that the doubling technique works for both the ElGamal scheme and the Goldwasser-Micali scheme.

Theorem 9. *Let $S = (G, E, D)$ be the ElGamal scheme. Then the scheme $S' = (G, E', D')$ obtained from S by the doubling technique is both IND-CPA and CUD-CPA secure under the DDH assumption.*

Proof. Let $\overline{\mathcal{C}}' = \mathbb{G}^4$, where $\mathbb{G}$ is the group where the computations are performed. Then the properties 1 and 2 of Definition 1 are satisfied. Moreover, a quadruple (u_1, v_1, u_2, v_2) is a valid ciphertext in S', if and only if $(g, u_1/u_2, y, v_1/v_2)$ is a DDH quadruple. Hence, assuming the intractability of the DDH problem, S' is CUD-CPA secure.

In order to prove that S' is IND-CPA secure, suppose there exists a successful adversary A' for S' in the IND-CPA game. Then there exists a successful attacker A on the IND-CPA security of S. First, A simulates A' and returns m_0, m_1 chosen by A'. After receiving the challenge ciphertext $c = E_{pk}(m_b)$, A uses the ciphertext self-reducibility property of ElGamal, and produces the following challenge for A': $c' = c \,||\, R(c)$. The answer of A' is used as an answer by A. Clearly, the ciphertext c' is a valid ciphertext in S'. Thus, the advantage of A is equal to the advantage of A'. □

Theorem 10. *Let $S = (G, E, D)$ be the Goldwasser-Micali scheme. Then the scheme $S' = (G, E', D')$ obtained from S by the doubling technique is both IND-CPA and CUD-CPA secure under the QR assumption.*

Proof. Let $\overline{\mathcal{C}}' = (\mathbb{Z}_n^*)^2$. Then the properties 1 and 2 of Definition 1 are obviously satisfied. Moreover, a pair (c_1, c_2) is a valid ciphertext in S', if and only if both c_1 and c_2 are either squares or pseudo-squares. Hence, assuming the intractability of distinguishing squares from pseudo-squares, S' is CUD-CPA secure.

The proof that S' is IND-CPA secure is analogous to the proof of Theorem 9 (using ciphertext self-reducibility of Goldwasser-Micali scheme). □

Theorems 9 and 10 give a constructive proof that using two independent encryptions of the plaintext in some IND-CPA secure scheme results in a CUD-CPA secure scheme. The proofs can be easily generalized for any encryption scheme that has the ciphertext self-reducibility property, satisfies $E(\mathcal{P}) = \mathcal{C}_v$ and this set is easily recognizable.

Doubling technique has mostly theoretical value. For practical purposes, doubling the size of the ciphertext is hardly acceptable.

5.2 Padding

Padding is another technique for producing CUD-secure schemes while preserving their IND-security. In contrast to the doubling, it is more general and much more practical. Let $S = (G, E, D)$ be an public-key encryption scheme. Corresponding scheme with a padding, $S' = (G, E', D')$, is defined as follows:

```
Algorithm E'_pk(m)
  return E_pk(0 || m)
```

```
Algorithm D'_sk(c)
  parse D_sk(c) as b || m
  if b = 0 then return m
  return ⊥
```

Notice that the plaintexts are one bit shorter in the new scheme. In addition, we shall require that the set of all valid ciphertext $\mathcal{C}_v$ in S can be easily recognized.

Theorem 11. *Let S be an IND-*atk *secure scheme, for atk $\in \{CPA, CCA1, CCA2\}$. Let the set of all valid ciphertexts in S, $\mathcal{C}_v$, be easily recognizable in probabilistic polynomial time. Then the scheme S' obtained from S by using padding is CUD-atk secure, and IND-atk secure as well.*

Proof. Let us start with the security in the CPA model, i.e. *atk*=CPA. First, let us assure that S' is IND-CPA secure. Let A' be a successful attacker on IND-CPA in S'. Then we can build an attacker A for S as follows: A simulates A' in the find phase and transforms the obtained plaintexts m'_0, m'_1 into his own plaintexts $m_0 = 0 \,||\, m'_0$, $m_1 = 0 \,||\, m'_1$. After receiving the challenge $c = E_{pk}(m_b)$, A passes c to A'. Whatever A' outputs, so does A. Since c is a valid ciphertext of m_b in S', the advantages of A and A' are equal.

Let us define $\overline{\mathcal{C}'} = \mathcal{C}_v$ (the properties 1 and 2 of Definition 1 are easily satisfied). In order to prove the CUD-CPA security of S', we show how a successful attacker A' on CUD-CPA security of S' can be turned into a successful attacker A on IND-CPA security of S. First, A chooses two plaintexts $m_0 = 0 \,||\, m'_0$, $m_1 = 1 \,||\, m'_1$, where m'_0 and m'_1 are chosen randomly. After receiving the challenge $c = E_{pk}(m_b)$, A passes it to A' as a challenge in the CUD-CPA game. The ciphertext c is valid in S' if and only if c is encryption of m_0 in S. Therefore, A can use the answer of A' to distinguish encryption of m_0 from encryption of m_1 with the same advantage as A' distinguished valid and invalid ciphertexts.

The proof for the CCA1 and CCA2 attack models is similar to the previous case. This is because the attacker A can simulate the decryption oracle for A' by using its own decryption oracle $D(.)$ and following the definition of $D'(.)$. In the CCA2 scenario, the same queries are prohibited in both games, hence the simulation is perfect. □

5.3 Schemes with the CUD Property

Another modification of the basic ElGamal encryption scheme that achieves ciphertext undetectability (even CUD-CCA1) is the Damgård ElGamal scheme [7] known as DEG. The scheme is described in Fig. 1.

Theorem 12. *The DEG encryption scheme is CUD-CPA secure under the DDH assumption.*

Proof. We shall prove that the set $\overline{\mathcal{C}} = \mathbb{G}^3$ has the required properties. Let A be a successful CUD-CPA attacker on DEG, we shall use it to construct a DDH distinguisher A' with the same advantage. Let $(s_1, s_2, t_1, t_2) \in \mathbb{G}^4$ be the DDH challenge, i.e. the task of A' is to determine whether $\log_{s_1} t_1 = \log_{s_2} t_2$. It picks $X_1, U \stackrel{R}{\leftarrow} \mathbb{G}$ at random and runs A as a subroutine, supplying it with the public key $pk = (s_1, s_2, X_2)$ and the challenge $c^* = (t_1, t_2, U)$. The distribution of (s_1, s_2, t_1, t_2) and the random choice of the values X_1 and U imply that for A this is a perfect simulation of the CUD experiment. It is easy to see that c^* is a valid ciphertext precisely if (s_1, s_2, t_1, t_2) is a DDH tuple, hence the advantages of A and A' are the same. □

Bellare and Palacio [4] have proved that the DEG scheme also achieves the PA1 form of plaintext awareness under another assumption posed on the group $\mathbb{G}$ beside the DDH assumption. They formalize this assumption (originally stated by Damgård in [7]) under the name Diffie-Hellman Knowledge (DHK1) assumption. Details of the formalization can be found in [4]. Their result together with Theorem 6 imply the following corollary.

Corollary 1. *The DEG encryption scheme is CUD-CCA1 secure under the DDH and DHK1 assumptions.*

Let us now consider the CS-lite encryption scheme defined in [6]. This is a simplification of the original Cramer-Shoup scheme [5] designed to achieve non-adaptive chosen ciphertext security. The scheme is described in Fig. 2.

Theorem 13. *The CS-lite encryption scheme is CUD-CPA secure under the DDH assumption.*

Proof. We shall prove that the set $\overline{\mathcal{C}} = \mathbb{G}^4$ has the properties required by Definition 1. It obviously satisfies properties 1 and 2, it remains to prove the third property. Let $(g_1, g_2, x_1, x_2, z) \in \mathbb{Z}_q^5$ be a random fixed private key. Let us partition the set $\overline{\mathcal{C}}$ into four subsets as follows:

$$\begin{aligned}
K &= \{(R_1, R_2, E, V) \in \mathbb{G}^4 \mid \log_{g_1} R_1 = \log_{g_2} R_2 \wedge V = R_1^{x_1} R_2^{x_2}\} \\
L &= \{(R_1, R_2, E, V) \in \mathbb{G}^4 \mid \log_{g_1} R_1 \neq \log_{g_2} R_2 \wedge V = R_1^{x_1} R_2^{x_2}\} \\
M &= \{(R_1, R_2, E, V) \in \mathbb{G}^4 \mid \log_{g_1} R_1 = \log_{g_2} R_2 \wedge V \neq R_1^{x_1} R_2^{x_2}\} \\
N &= \{(R_1, R_2, E, V) \in \mathbb{G}^4 \mid \log_{g_1} R_1 \neq \log_{g_2} R_2 \wedge V \neq R_1^{x_1} R_2^{x_2}\}
\end{aligned}$$

To finish the proof, we need to show that $K \cup L \approx_c M \cup N$, assuming DDH.

First, let us prove that $K \approx_c M$ under the DDH assumption. Suppose there exists a probabilistic polynomial algorithm A that distinguishes the sets K and M with a non-negligible advantage. We shall use A to construct a DDH distinguisher A'. Let (s_1, s_2, t_1, t_2) be the challenge for A', i.e. its task is to decide

```
Algorithm G(1^k)
  choose
    k-bit prime q
    group G of order q
  g ←R G
  x1, x2 ←R Zq
  X1 ← g^x1
  X2 ← g^x2
  pk ← (g, X1, X2)
  sk ← (g, x1, x2)
  return (pk, sk)
```

```
Algorithm E_pk(m)
  y ←R Zq
  Y ← g^y
  W ← X1^y
  U ← m · X2^y
  return (Y, W, U)

Algorithm D_sk(Y, W, U)
  if W = Y^x1
    then return U · Y^(−x2)
    else return ⊥
```

Fig. 1. The DEG encryption scheme.

```
Algorithm G(1^k)
  choose
    k-bit prime q
    group 𝔾 of order q
  g1, g2 ←R 𝔾
  x1, x2, z ←R ℤ_q
  X ← g1^x1 g2^x2
  Z ← g1^z
  pk ← (g1, g2, X, Z)
  sk ← (g1, g2, x1, x2, z)
  return (pk, sk)

Algorithm E_pk(m)
  r ←R ℤ_q
  R1 ← g1^r
  R2 ← g2^r
  E ← m · Z^r
  V ← X^r
  return (R1, R2, E, V)

Algorithm D_sk(R1, R2, E, V)
  if V = R1^x1 R2^x2
    then return E/R1^z
    else return ⊥
```

Fig. 2. The CS-lite encryption scheme.

whether $\log_{s_1} t_1 = \log_{s_2} t_2$. First, A' picks at random $w \xleftarrow{R} \mathbb{Z}_q$ and $E, Z \xleftarrow{R} \mathbb{G}$. Then it determines the following values:

$$\begin{array}{lll} g_1 := s_1 & R_1 := s_2 & X := t_1 \\ g_2 := s_1^w & R_2 := s_2^w & V := t_2 \end{array}$$

Finally, A' runs A as a subroutine with the public key (g_1, g_2, X, Z) and the challenge (R_1, R_2, E, V) and returns the same answer A does. Since $\log_{g_1} u_1 = \log_{g_2} u_2 = \log_{s_1} s_2$ (let us denote this value r), the challenge ciphertext belongs to $K \cup M$ and for A, this is a perfect simulation of the computational distinguishability game. To see that the advantages of A and A' are the same, note that $(R_1, R_2, E, V) \in K \Leftrightarrow V = R_1^{x_1} R_2^{x_2}$. Since $\log_{g_1} u_1 = \log_{g_2} u_2$, this is equivalent to $V = g_1^{r(x_1 + w x_2}$. This equation holds if and only if (g_1, R_1, X, V) is a DDH tuple. But this is just a different notation for the tuple (s_1, s_2, t_1, t_2).

By a similar argument, it can be proved that $K \approx_c L$ and $M \approx_c N$ under the DDH assumption. In both cases, any successful distinguisher A can be transformed into a DDH distinguisher. To decide whether (s_1, s_2, t_1, t_2) is a DDH tuple, it suffices to query A for a ciphertext with $R_1 := t_1$ and $R_2 := t_2$, providing a public key where $g_1 := s_1$ and $g_2 := s_2$. The rest of the public key and the challenge ciphertext can be easily chosen to provide a perfect simulation for A.

Since $K \approx_c L$, $K \approx_c M$ and $M \approx_c N$, the difference in behavior of any probabilistic polynomial algorithm on these sets must be negligible. Therefore, it also holds $K \cup L \approx_c M \cup N$, which proves the CUD-CPA security of the CS-lite encryption scheme under the DDH assumption. □

Similar to the DEG scheme, CS-lite was also proved to be PA1 under the DDH and DHK1 assumptions by Bellare and Palacio [4]. Hence we can again use Theorem 6 to obtain the following corollary.

Corollary 2. *The CS-lite encryption scheme is CUD-CCA1 secure under the DDH and DHK1 assumptions.*

```
Algorithm G(1^k)
  choose
    k-bit prime q
    group 𝔾 of order q
    H : 𝔾^3 → ℤ_q
  g_1, g_2 ←R 𝔾
  x_1, x_2, y_1, y_2, z_1, z_2 ←R ℤ_q
  c ← g_1^{x_1} g_2^{x_2}
  d ← g_1^{y_1} g_2^{y_2}
  h ← g_1^{z_1} g_2^{z_2}
  pk ← (g_1, g_2, c, d, h, H)
  sk ← (x_1, x_2, y_1, y_2, z_1, z_2)
  return (pk, sk)
```

```
Algorithm E_pk(m)
  r ←R ℤ_q
  u_1 ← g_1^r
  u_2 ← g_2^r
  e ← m · h^r
  α ← H(u_1, u_2, e)
  v ← v^r · d^{rα}
  return (u_1, u_2, e, v)

Algorithm D_sk(u_1, u_2, e, v)
  α ← H(u_1, u_2, e)
  if v = u_1^{x_1+y_1α} u_2^{x_2+y_2α}
    then return e/(u_1^{z_1} u_2^{z_2})
    else return ⊥
```

Fig. 3. The Cramer-Shoup encryption scheme.

Finally, let us consider the ciphertext undetectability properties of the original Cramer-Shoup encryption scheme defined in [5]. Recall the definition of the original scheme, depicted in Fig. 3.

Theorem 14. *The Cramer-Shoup encryption scheme is CUD-CPA secure under the DDH assumption.*

Proof. The proof is analogous to the proof of Theorem 13. Again we choose the set $\overline{\mathcal{C}} = \mathbb{G}^4$, pick a random private key $(x_1, x_2, y_1, y_2, z_1, z_2) \in \mathbb{Z}_q^6$ and partition the set $\overline{\mathcal{C}}$ into four subsets as follows:

$$K = \{(u_1, u_2, e, v) \in \mathbb{G}^4 \mid \log_{g_1} u_1 = \log_{g_2} u_2 \wedge v = u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha}\}$$
$$L = \{(u_1, u_2, e, v) \in \mathbb{G}^4 \mid \log_{g_1} u_1 \neq \log_{g_2} u_2 \wedge v = u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha}\}$$
$$M = \{(u_1, u_2, e, v) \in \mathbb{G}^4 \mid \log_{g_1} u_1 = \log_{g_2} u_2 \wedge v \neq u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha}\}$$
$$N = \{(u_1, u_2, e, v) \in \mathbb{G}^4 \mid \log_{g_1} u_1 \neq \log_{g_2} u_2 \wedge v \neq u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha}\}$$

Here α denotes the value $H(u_1, u_2, e)$. To finish the proof, we again need to show that $K \cup L \approx_c M \cup N$, assuming DDH.

The only part that differs from the proof of Theorem 13 is to show that $K \approx_c M$ under the DDH assumption. Let us consider the same setting as in the previous proof and show how A' can use A to help it solve the DDH problem. First, A' picks at random $w, y_1, y_2, z_1, z_2 \xleftarrow{R} \mathbb{Z}_q$ and $e \xleftarrow{R} \mathbb{G}$. Then it determines the following values:

$$g_1 := s_1 \qquad h := g_1^{z_1} g_2^{z_2} \qquad v := t_2$$
$$g_2 := s_1^w \qquad u_1 := s_2 \qquad \alpha := H(u_1, u_2, e)$$
$$d := g_1^{y_1} g_2^{y_2} \qquad u_2 := s_2^w \qquad c := t_1/d^\alpha$$

Finally, A' runs A as a subroutine with the public key (g_1, g_2, c, d, h, H) and the challenge ciphertext (u_1, u_2, e, v) and returns the same answer A does. Since $\log_{g_1} u_1 = \log_{g_2} u_2 = w$, the challenge ciphertext belongs to $K \cup M$ and for A, this is again a perfect simulation of the computational distinguishability game. The advantages of A and A' are the same, because $(u_1, u_2, e, v) \in K \Leftrightarrow v = u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha}$. Since $\log_{g_1} u_1 = \log_{g_2} u_2$, this is equivalent to $v = c^r d^{r\alpha}$, where r denotes the value $\log_{g_1} u_1$. This equation holds if and only if $(g_1, g_1^r, cd^\alpha, v)$ is a DDH tuple, which corresponds to the tuple (s_1, s_2, t_1, t_2). The rest of the argument is analogous to the proof of Theorem 13. □

6 Conclusion and Further Research

In this paper, we have explored the idea of ciphertext undetectability. We have compared this notion with the indistinguishability of ciphertexts and plaintext awareness. We have proved that, in general, these notions are unrelated. Since ciphertext undetectability does not provide secrecy for public-key schemes, it is important to find constructions that turn IND-*atk* or PA secure schemes into schemes having CUD property while preserving their IND-*atk* or PA security. The constructions given in Section 5, namely the doubling technique and padding, can be applied only on schemes satisfying additional properties (e.g. cipher self-reducibility or easily recognizable set $\mathcal{C}_v$). Definitely, more general constructions, without additional requirements, are needed.

Other approach to obtain schemes with the CUD property is to prove the CUD-*atk* security of some established schemes directly. Although we have proved the original Cramer-Shoup encryption scheme to be CUD-CPA secure, we conjecture it to achieve the CUD-CCA2 degree of security. However, proving this would require a more sophisticated approach than the approach used in proving the CUD-CCA1 security of CS-lite. This is mainly because the PA1 form of plaintext awareness achieved by CS-lite plays a useful role also for ciphertext undetectability (cf. Theorem 6, but this does not seem to be true for the stronger PA2 notion. In other words, the implication PA2 + CUD-CPA ⇒ CUD-CCA2 does not seem to hold. This is because the PA2 notion was designed to model also the possibility of the adversary to get hold of one ciphertext (the challenge) for which it does not know the corresponding plaintext. However, in the CUD experiment, it can also receive an invalid "ciphertext", so some modifications of the model would be needed.

References

1. Ahn, L., Hopper, N.: Public-Key Steganography, Advances in Cryptology – Eurocrypt 2004, LNCS vol. 3027, pp. 322–339, Springer, 2004.
2. Backes, M., Cachin, C.: Public-Key Steganography with Active Attacks, Cryptology ePrint Archive, Report 2003/231, `http://eprint.iacr.org/2003/231`, 2003.
3. Bellare, M., Desai, A., Pointcheval, D. and Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes, Advances in Cryptology – Crypto 98 Proc., LNCS vol. 1462, H. Krawzyk ed, Springer-Verlag, 1998

4. Bellare, M. and Palacio, A.: Towards Plaintext-Aware Public-Key Encryption Without Random Oracles, ASIACRYPT 2004 Proc., LNCS vol. 3329, pp. 48–62, Springer-Verlag, 2004
5. Cramer, R. and Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, Advances in Cryptology – Crypto 98 Proc., LNCS vol. 1462, H. Krawzyk ed, Springer-Verlag, 1998
6. Cramer, R. and Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack, SIAM Journal on Computing vol. 33, pp. 167–226, 2003
7. Damgård, I: Towards practical public key systems secure against chosen ciphertext attacks, Advances in Cryptology – Crypto 91 Proc., LNCS vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991
8. Dent, A. W.: The Cramer-Shoup Encryption Scheme is Plaintext Aware in the Standard Model, Advances in Cryptology – Eurocrypt 2006 Proc., LNCS vol. 4004, pp. 289–307, S. Vaudenay ed, Springer-Verlag, 2006
9. Hall, C., Goldberg, I., and Schneier, B.: Reaction Attacks against several Public-Key Cryptosystems, Information and Communication Security, Second International Conference, ICICS'99, Springer-Verlag, LNCS vol. 1726, pp. 2–12, 1999.
10. Möller, B.: A Public-Key Encryption Scheme with Pseudo-Random Ciphertexts. ESORICS 2004, LNCS vol. 3193, pp. 335-351, Springer-Verlag, 2004.
11. Pointcheval, D.: Contemporary Cryptology – Provable Security for Public Key Schemes, Advanced Courses CRM Barcelona, pp. 133–189, 2005.