Relaxing IND-CCA: Indistinguishability Against Chosen Ciphertext *Verification* Attack

No Author Given

No Institute Given

Abstract

The definition of IND-CCA security model for public key encryption allows an adversary to obtain (adaptively) decryption of ciphertexts of its choice. That is, the adversary is given oracle access to the decryption function corresponding to the decryption key in use. The adversary may make queries that do not correspond to a *valid* ciphertext, and the answer will be accordingly (i.e., a special "failure" symbol).

In this article, we investigate the case where we restrict the oracle to only determine if the query made is a valid ciphertext or not. That is, the oracle will output 1 if the query string is a valid ciphertext (do not output the corresponding plaintext) and output 0 otherwise. We call this oracle as "*ciphertext verification oracle*" and the corresponding security model as Indistinguishability against *chosen ciphertext verification attack* (IND-CCVA). We point out that this seemingly weaker security model is meaningful, clear and useful to the extent where we motivate that certain cryptographic functionalities can be achieved by ensuring the IND-CCVA security where as IND-CPA is not sufficient and IND-CCA provides more than necessary. We support our claim by providing nontrivial construction (existing/new) of:

- public key encryption schemes that are IND-CCVA secure but not IND-CCA secure,
- public key encryption schemes that are IND-CPA secure but not IND-CCVA secure.

Our discoveries are another manifestation of the subtleties that make the study of security notions for public key encryption schemes so attractive and are important towards achieving the definitional clarity of the target security.

Keywords: PKE security notions, IND-CPA, IND-CCA.

1 Introduction

The IND-CCA security (security against adaptive chosen ciphertext attacks [15, 19, 3, 6]) is now a days considered the *de facto* level of security required for public key encryption schemes used in practice. Unfortunately, only a handful of approaches are known for constructing encryption schemes that meet this notion of security, thus showing the strongness of this security model. In practice, there are certain cryptographic functionalities for which the security requirement is apparently **less stronger** than IND-CCA. There had been some research to quantify the gap between the IND-CPA and IND-CCA security.

The most common threat to IND-CCA security is that of a query on a malformed ciphertext causing the decryption oracle to leak damaging information, either about the private key, or about the plaintext. Understanding, the explicit behaviour of the decryption oracle could be the keypoint. In IND-CCA model, the decryption oracle provides decryption of the ciphertexts of our choice. In this work, we limit the output of the decryption oracle: it will only verify whether or not a query string is a valid ciphertext or not. We will show that this seemingly weaker security model is meaningful.

1.1 Background

The security for public key encryption was first formally defined by Goldwasser and Micali [11]. Their notion of *semantic security*, roughly speaking, requires that observation of a ciphertext does not enable an adversary to compute anything about the underlying plaintext message that it could not have computed on its own (i.e., prior to observing the ciphertext). Goldwasser and Micali (see also [9, 8, 9]) proved that semantic security is equivalent to the notion of *indistinguishability* that requires (roughly) the following: given a public key pk, a ciphertext C, and two possible plaintexts m_0, m_1, it is infeasible to determine if C is an encryption of m_0 or an encryption of m_1. We will refer to these notions using the commonly accepted term "IND-CPA" security.

IND-CPA security does not guarantee any security against *chosen ciphertext* attacks by which an adversary may obtain decryption of ciphertexts of its choice. Indistinguishability based definitions appropriate for this setting were given by Naor and Yung [15] and Rackoff and Simon [19]. Naor and Yung consider *non-adaptive* chosen ciphertext attack in which the adversary may request decryptions only *before* it obtains the challenge ciphertext. Rackoff and Simon define the stronger notion of security against *adaptive* chosen ciphertext attacks whereby the adversary may request decryptions even after seeing the challenge ciphertext, under the natural limitation that the adversary may not request decryption of the challenge ciphertext it self. We will refer to the later notion as "IND-CCA" security.

1.2 Summary of Our Results

In the definition of IND-CCA security model for public key encryption, the adversary is given oracle access to the decryption function corresponding to the decryption key in use. The adversary may make queries that do not correspond to a *valid* ciphertext, and the answer will be accordingly (i.e., a special "failure" symbol). In this article, we investigate the case where we restrict the oracle to only verify if the query made is a valid ciphertext or not. That is, the oracle will output 1 (not the corresponding plaintext) if the query string is a valid ciphertext and output 0 otherwise. We will denote this oracle by the name "*ciphertext verification oracle*" and the corresponding security model by the name Indistinguishability against *chosen ciphertext verification attack* (IND-CCVA). We point out that this seemingly weaker security model is meaningful, clear and useful since by observing certain results from cryptographic literature [13], it is clear that certain cryptographic functionalities can be achieved by ensuring the IND-CCVA security, where IND-CPA is not sufficient and IND-CCA provides more than necessary. We further support our claim by providing nontrivial construction of:

- public key encryption schemes that are IND-CCVA secure but not IND-CCA secure,
- public key encryption schemes that are IND-CPA secure but not IND-CCVA secure.

1.3 Organization

In the following section, we fix notations, recall some notations from number theory, and provide an informal overview of public key encryption and the IND-CCA security model. Formal definition of IND-CCVA security model appear in Section 3. The first of our separating schemes is discussed in Section 4. The public key encryption discussed in this section is IND-CCVA secure but not IND-CCA. This scheme, thus separate the IND-CCVA and IND-CCA security models.

We show hope in IND-CCVA security by looking at a existing cryptographic functionality (requirement) which can not be achieved by ensuring IND-CPA security and IND-CCA security guarantee add to redundancy: we show infact this requirement is exactly fulfilled by ensuring IND-CCVA security. The above is discussed in Section 5 with the supporting separating schemes (IND-CPA secure but not IND-CCVA): existing and a new proposal.

2 Preliminaries

In this section we first fix the notations. We write $x \xleftarrow{\mathcal{R}} X$ to denote the action of assigning a value to the variable x sampled uniformly from the set X. If $\mathcal{A}$ is a probabilistic algorithm which takes an input x, $\mathcal{A}(x)$ denotes the output distribution of $\mathcal{A}$ on input x. Hence, $y \xleftarrow{\mathcal{R}} \mathcal{A}(x)$ denotes the assignment of a value to the variable y from the output distribution of algorithm $\mathcal{A}$ on input x. We can denote any probabilistic algorithm $\mathcal{A}$ by a deterministic algorithm $\mathcal{A}'$ which takes additional input, random coins, uniformly sampled from some set R.

We now recall some well-known notations from number theory. Let n be a positive integer. The number of positive integers less than n and relatively prime to n is denoted $\phi(n)$. The ring of integers modulo n, $\mathbb{Z}_n$, is defined to be the set $\{0, \ldots, n-1\}$ equipped with modulo n operations: $+$ and $\times$. The set of residues modulo n that are relatively prime to n is denoted $\mathbb{Z}_n^*$. The set $\{a \in \mathbb{Z}_n^* :$ the Jacobi Symbol $(\frac{a}{n}) = 1\}$ is denoted $J(n)$. The set of *quadratic residues* $\{a \in \mathbb{Z}_n^* : x^2 \equiv a \pmod{n}$ is solvable$\}$ is denoted $QR(n)$.

We take any string x as a $\{0,1\}$-string. If u and v are two strings, then uv or $u||v$ denotes the concatenation of strings u and v. $|x|$ denotes the length of the string x. $lsb(x)$ and $msb(x)$ denotes the least and the most significant bit of string x respectively. We denote the least k significant bits of string x by $lsb(x,k)$. For a number x, the $lsb(x)$ denotes the least significant bit of the corresponding string representation of x. Similar interpretation for $msb(x)$, $lsb(x,k)$, and $|x|$ when x is a number.

PKE. A public key encryption scheme $\prod=$ (KeyGen, Enc, Dec) is a triple of algorithms. The key generation algorithm KeyGen takes a security parameter 1^λ and returns a pair (pk, sk) of matching public and secret keys. The encryption algorithm Enc takes a public key pk and a message $m \in \{0,1\}^*$ to produce a ciphertext C. The deterministic decryption algorithm Dec takes sk and ciphertext C to produce either a message $m \in \{0,1\}^*$ or a special symbol $\perp$ to indicate that the ciphertext was invalid. The consistency requirement is that for all $\lambda \in \mathbb{N}$, for all (pk, sk) which can be output by KeyGen(1^λ), for all $m \in \{0,1\}^*$ and for all C that can be output by Enc(pk, m), we have that Dec$(sk, C) = m$.

IND-CCA. We recall the definitional template of the IND-CAA security model. The underlying experiment picks a public key pk and matching secret key sk, and then provides pk to the adversary $\mathcal{A}$. The latter runs in two phases in both of which $\mathcal{A}$ has access to an oracle for decryption under sk. It ends its first phase by outputting a pair m_0, m_1 of messages. The experiment picks a challenge bit $b \in \{0,1\}$ at random, encrypts m_b under pk, and returns the resulting challenge ciphertext C^* to $\mathcal{A}$. The latter now enters its second phase, which it ends by outputting a bit $\bar{b}$. We say that $\mathcal{A}$ wins if $b = \bar{b}$. Security requires that the probability of winning minus $\frac{1}{2}$ is negligible.

3 IND-CCVA: Indistinguishability against Chosen Ciphertext Verification Attack

We now present a formal definition of security against chosen ciphertext verification attacks. This is a weaker form of attack when compared to a full CCA attack: the adversary has access to a oracle which is weaker than a decryption oracle. We name this oracle as ciphertext verification oracle and denoted it by $\mathcal{O}_{CV}$. The oracle is described as follows:

$$\mathcal{O}_{CV} : \{0,1\}^* \rightarrow \{0,1\}$$

The output is 1 if and only if the input string is a *valid ciphertext*. We now describe this new attack model formally as follows. For a public key encryption scheme $\prod$ and an adversary $\mathcal{A}$, consider the following experiment:

The IND-CCVA Experiment

- KeyGen(1^λ) is run to obtain keys (pk, sk).

– Beside the public key pk, the adversary $\mathcal{A}$ is given access to ciphertext verification oracle $\mathcal{O}_{CV}$.
– The adversary outputs a pair of messages m_0, m_1 of the same length from the plaintext space.
– A random bit $b \leftarrow \{0,1\}$ is chosen, and then a ciphertext $c \leftarrow \text{Enc}_{pk}(m_b)$ is computed and given to $\mathcal{A}$.
– $\mathcal{A}$ continues to interact with $\mathcal{O}_{CV}$.
– Finally, $\mathcal{A}$ outputs a bit $\bar{b}$.
– The output of the experiment is defined to be 1 if $\bar{b} = b$, and 0 otherwise.

We define the advantage of $\mathcal{A}$ in the IND-CCVA experiment as a function of the security parameter as follows:

$$\text{Adv}^{IND-CCVA}_{\prod,\mathcal{A}}(\lambda) \stackrel{\Delta}{=} |\text{Prob}[\bar{b} = b] - \frac{1}{2}| \tag{1}$$

Definition 1. *A public key encryption scheme $\prod$=(KeyGen, Enc, Dec) has indistinguishable encryption under a decisional chosen ciphertext attack (or is IND-CCVA secure) if for all probabilistic polynomial time adversaries $\mathcal{A}$, we have that $Adv^{IND-CCVA}_{\prod,\mathcal{A}}(\lambda)$ is negligible.*

Note that our oracle may become constant (always output 1) for certain class of public key encryption schemes. Let $\prod$ be a public key encryption scheme with $\mathcal{K}$ as key space, $\mathcal{M}$ as message space, and $\mathcal{C}$ as ciphertext space. In general, we have

$$\cup_{k\in\mathcal{K}}\text{Enc}(\mathcal{M}) \subsetneq \mathcal{C}.$$

The equality between $\cup_{k\in\mathcal{K}}\text{Enc}(\mathcal{M})$ and $\mathcal{C}$ means that any element from $\mathcal{C}$ is a valid ciphertext (encryption of some message under some key). Thus in this case, the verification oracle will always output 1 for any random query from $\mathcal{C}$. This will imply that the IND-CPA and IND-CCVA security are both equivalent for such public key encryption schemes (as oracle is of no use).

Thus, in this article we consider public keys encryption schemes with $\cup_{k\in\mathcal{K}}\text{Enc}(\mathcal{M}) \subsetneq \mathcal{C}$. Infact, achieving IND-CCA security requires this kind of setup in general.

4 The Separating Scheme: IND-CCVA secure but not IND-CCA secure

In this section we describe a public key encryption scheme which was originally proposed by Cramer and Shoup [5] as the light version of their main scheme (the first practical IND-CCA secure scheme). The scheme was shown to be IND-CPA secure by Cramer and Shoup and not IND-CCA secure. We observed that this scheme is infact IND-CCVA secure, thus settling the claim of this section.

4.1 Cramer-Shoup light version

– **KG(1^λ):** The key generation algorithm runs as follows.
  • Choose a group G of prime order p, where $2^{\lambda-1} < p < 2^\lambda$
  • Choose $g_1, g_2 \stackrel{\mathcal{R}}{\leftarrow} G$ and $x_1, x_2, z \in \mathbb{Z}_p$.
  • Compute $c = g_1^{x_1} g_2^{x_2}$ and $h = g_1^z$.
  • The public key, PK, for this scheme is tuple (g_1, g_2, c, h), with corresponding secret key, SK, is (x_1, x_2, z).
  • message space = G.
– **ENC(m, PK):** To encrypt a message $m \in G$, the encryption algorithm runs as follows.
  • Choose $r \stackrel{\mathcal{R}}{\leftarrow} \mathbb{Z}_p$.
  • Compute $u_1 = g_1^r,\ u_2 = g_2^r,\ e = h^r m,\ v = c^r$.
  • The ciphertext, $\mathcal{C}$, is (u_1, u_2, e, v).
– **DEC($\mathcal{C}, SK, PK$):** Decryption works in the following way: given the ciphertext (u_1, u_2, e, v) and secret key (x_1, x_2, z),

- It first tests if $u_1^{x_1} u_2^{x_2} \stackrel{?}{=} v$.
- If this condition does not hold, the decryption algorithm outputs $\perp$; otherwise, it outputs

$$m = \frac{e}{u_1^z}.$$

Correctness. If (u_1, u_2, e, v) is a valid ciphertext, then we have:

$$u_1^{x_1} u_2^{x_2} = g_1^{r x_1} g_2^{r x_2} = g_1^{x_1} g_2^{x_2 r} = c^r = v \text{ and}$$

$$\frac{e}{u_1^z} = \frac{h^r m^{\,z}}{g_1^r} = \frac{g_1^{z r} m}{g_1^{rz}} = \frac{g_1^{rz} m}{g_1^{rz}} = m.$$

4.2 IND-CCVA Security

We show that this scheme is IND-CCVA secure based on the hardness of the Decisional Diffie-Hellman (DDH) problem in G.

DDH problem can be formulated as follows. Let $\mathcal{D}$ be an algorithm that takes triples of group elements as input and outputs a bit. The DDH-advantage of $\mathcal{D}$ is defined as

$$\left| \Pr[x, y \stackrel{\mathcal{R}}{\leftarrow} G : \mathcal{D}(g^x, g^y, g^{xy}) = 1] - \Pr[x, y, z \stackrel{\mathcal{R}}{\leftarrow} G : \mathcal{D}(g^x, g^y, g^z) = 1] \right|$$

Then DDH assumption for G assumes that for any efficient algorithm $\mathcal{D}$, it's DDH-advantage is negligible.

Theorem 1. *The scheme described in Section 4.1 is IND-CCVA secure assuming that the DDH assumption holds in G.*

Proof. The proof goes by reduction which shows that if an adversary is able to break the IND-CCVA security, it can be used to solve the DDH problem. Let us assume, there is an adversary $\mathcal{A}$ which can break the IND-CCVA security of the scheme. Using $\mathcal{A}$, we can construct an algorithm $\mathcal{B}$ that solves the DDH problem.

$\mathcal{B}$ is given as input a 4-tuple (g, g^a, g^b, Z). The task of $\mathcal{B}$ is to determine whether Z is equal to g^{ab} or a random element of G. $\mathcal{B}$ solves this problem by interacting with $\mathcal{A}$ in the IND-CCVA game as follows.

– **Simulation of Key Generation (KG):** $\mathcal{B}$ proceeds as follows:
  - Sets $g_1 = g$.
  - Chooses $s \stackrel{\mathcal{R}}{\leftarrow} \mathbb{Z}_p$ and sets $g_2 = g_1^s$.
  - Chooses $x_1, x_2 \stackrel{\mathcal{R}}{\leftarrow} \mathbb{Z}_p$ and sets $c = g_1^{x_1} g_2^{x_2}$.
  - Sets $h = g^b$.
  - Finally the 4-tuple (g_1, g_2, c, h) is made available as public key to $\mathcal{A}$ by $\mathcal{B}$.

– **Simulation of Ciphertext Verification Oracle for Ciphertext Validity Check:**
  - Knowledge of (x_1, x_2) ensures that $\mathcal{B}$ can perfectly answer the ciphertext verification queries asked by $\mathcal{A}$.

– **Simulation of Challenge Ciphertext:**
  - In Challenge Phase, $\mathcal{A}$ chooses and outputs two messages m_0 and m_1 to $\mathcal{B}$.
  - $\mathcal{B}$ then chooses a bit $\tau \stackrel{\mathcal{R}}{\leftarrow} \{0, 1\}$ and it proceeds to encrypt m_τ.
  - $\mathcal{B}$ sets

$$u_1 = g^a,\ u_2 = (g^a)^s,\ e = Z \cdot m_\tau \text{ and } v = (g^a)^{x_1} (g^a)^{s x_2}.$$

- The challenge ciphertext (u_1, u_2, e, v) is given to $\mathcal{A}$ by $\mathcal{B}$.

Finally in the Guess Phase, $\mathcal{A}$ answers a bit τ'. If $\tau = \tau'$ then $\mathcal{B}$ announces the input instance to be a valid DDH tuple. This completes the description of $\mathcal{B}$. We show that

$$\mathcal{A}dv(\mathcal{B}) = \mathcal{A}dv(\mathcal{A}).$$

For this it is enough to show that simulation of challenge ciphertext is perfect given a valid DDH instance. This is true as for valid DDH tuple (i.e., $z = g^{ab}$) we have

- $u_1 = g^a = g_1^a$.
- $u_2 = (g^a)^s = (g^s)^a = g_2^a$.
- $e = Z \cdot m_\tau = g^{ab} \cdot m_\tau = (g^b)^a \cdot m_\tau = h^a \cdot m_\tau$.
- $v = (g^a)^{x_1}(g^a)^{sx_2} = (g^{x_1} g^{sx_2})^a = c^a$.

Thus the simulation of challenge ciphertext is perfect. This proves the theorem. □

Lemma 1. *The scheme described in Section 4.1 is not IND-CCA secure.*

Proof. In IND-CCA game, if $\mathcal{C} = (u_1, u_2, e, v)$ be the challenge ciphertext, adversary $\mathcal{A}$ chooses any message $m' \neq 1$ (identity in G) and creates another ciphertext $\mathcal{C}' = (u_1, u_2, m'e, v)$ which is indeed different than challenge ciphertext. Decryption oracle returns $m'm$ if $\mathcal{C}'$ is queried to it. $\mathcal{A}$ then easily calculates the original message by calculating $m'mm'^{-1} = m$. Hence, the lemma. □

5 The Separating Scheme (Known): IND-CPA secure but not IND-CCVA secure

It is well-known that plain-old RSA does not hide partial information about the plaintext, is malleable, and is also insecure against chosen ciphertext attack. Indeed, plain-old RSA is never used in practice, precisely because of these well-known weaknesses. Instead, what people actually use is plain-old RSA with a few modifications attempt to fix these problems.

One idea that is often advocated to improve the security of plain-old RSA is to use a randomized "encoding" or "padding" scheme. That is, we encrypt m as $C = f(m, r)^e$, where $f(m, r)$ encodes the message m using some random bits r. Note that f is not a cryptographic encoding: it is easy for anyone to compute m from $f(m, r)$. The hope is that this enhancement improves the security of RSA. However, if one is not extremely careful, the resulting scheme may become insecure.

One simple way to define $f(m, r)$ is just to concatenate the two bit strings m and r. This is a popular idea. RSA, Inc. has a very popular encryption function, called PKCS #1, which did essentially this until the well-known attack by Bleichenbacher [4] had surfaced. This encryption function is used by the security protocol SSL over internet.

In literature, Bleichenbacher's attack on SSL has been termed as chosen ciphertext attack on RSA's PKCS #1. But we observe that, his attack is actually a chosen ciphertext verification attack. We first describe briefly the RSA encryption standard PKCS #1; refer to [21] for details. It has three block formats: Block types 0 and 1 are reserved for digital signatures, and block type 2 is used for encryption. As we are interested in encryption only, we describe the block 2.

- **KG(1^λ):** Choose primes p, q ($4k$ bit each) and compute $n = pq$ (n is k byte number). Choose e, d, such that $ed \equiv 1 \pmod{\phi(n)}$. The public key, PK, is (n, e) and the secret key, SK, is (p, q, d).
- **ENC(m, PK):** A data block D, consisting of $|D|$ bytes, is encrypted as follows:
  - First, a padding string PS, consisting of $k - 3 - |D|$ nonzero bytes, is generated pseudo-randomly (the byte length of PS is atleast 8).

- Now, the encryption block $EB = 00||02||PS||00||D$ is formed, is converted into an integer x, and is encrypted with RSA, giving the ciphertext $c = x^e \pmod{n}$.

– **DEC**(c, SK, PK) A Ciphertext c is decrypted as follows:
- Compute $x' = c^d \pmod{n}$.
- Converts x' into an encryption block EB'.
- Check, if the encryption block is PKCS *conforming* (An encryption block EB consisting of k bytes, $EB = EB_1||\ldots||EB_k$, is called PKCS conforming, if it satisfies the following conditions: $EB_1 = 00$, $EB_2 = 02$, EB_3 through EB_{10} are nonzero and at least one of the bytes EB_{11} through EB_k is 00).
- If the encryption block is PKCS conforming, then output the data block; otherwise an error sign.

5.1 Security

It is well-known that the least significant bit of plain RSA encrypted message is as secure as the whole message [10, 1]. In particular, there exists an algorithm that can decrypt a ciphertext if there exists another algorithm that can predict the least significant bit of a message given only the corresponding ciphertext and the public key. Håstad and Näslund extended this result to show that all individual RSA bits are secure [12].

Bleichenbacher's attack assumes that the adversary has access to an oracle that, for every ciphertext, returns whether the corresponding plaintext is PKCS conforming. If the plaintext is not PKCS conforming, the oracle outputs an error sign. Given just these error signs, because of specific properties of PKCS #1, Bleichenbacher showed how a very clever program can decrypt a target ciphertext (the oracle answer will reveal the first two bytes of the corresponding plaintext of the chosen ciphertext). Though, at this point the algorithm of Håstad and Näslund can use this oracle to decrypt the target ciphertext, Bleichenbacher's attack, different from Håstad and Näslund, was aimed at minimizing the number of oracle queries; thus, showing the practicality of the attack.

Hence, all the attacker needs is the verification about the validity of the chosen ciphertext (and not the corresponding whole plaintext). Thus this is clearly a chosen ciphertext verification attack.

5.2 The Separating Scheme (New Proposal): IND-CPA secure but not IND-CCVA secure

In this section, we present an efficient IND-CPA secure public key encryption scheme $\mathcal{E}$. We show that this scheme is not secure against IND-CCVA attack. Theoretically this confirms the gap between IND-CPA and IND-CCVA attacks through a practical instantiation of public key encryption.

Here we reserve three symbols: n as a product of two prime numbers, $k = \lceil \log_2(\log_2(n)) \rceil$, and $l = |n| - k$ ($|n|$ denotes the bit-length of n).

5.3 Encryption Scheme $\mathcal{E}$

1. **KG**(1^λ)
   – Choose two odd prime numbers $2^{\lambda-1} < p' < q' < 2^\lambda$ such that $p = 2p' + 1$ and $q = 2q' + 1$, where p and q are also prime numbers.
   – Compute $n = pq$.
   – Compute $\phi(n) = (p-1)(q-1) = 4p'q'$.
   – Choose $e \in \mathbb{Z}_n$ such that $\gcd(e, \phi(n)) = 1$.
   – Compute d such that $ed \equiv 1 \pmod{\phi(n)}$
   – Compute $u = \frac{\phi(n)+4}{8}$

Public Key $PK = (n, e)$
Secret Key $SK = (p, q, d, u)$
Message Space $\mathcal{M} = J(n) \setminus \{-1, 1\}$
Ciphertext Space $\mathcal{C} = \mathbb{Z}_n \times J(n)$

2. **ENC**(m, PK)
   - Choose $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$
   - Let $\bar{m} = lsb(m, k)$
   - Compute $r = \bar{r}||\bar{m}$
   - Compute $c_1 \equiv r^e \pmod{n}$
   - Compute $c_2 \equiv m^{\bar{r}} \pmod{n}$

   Ciphertext $\mathcal{C} = (c_1, c_2)$
3. **DEC**(C, SK, PK)
   - Compute $r' = c_1^d \pmod{n}$
   - Let $r' = \bar{r}'||\bar{m}'$ where $|\bar{m}'| = k$
   - Let $\bar{r}' = 2^t s$, where s is an odd number.
   - Compute s' such that $s's \equiv 1 \pmod{\phi(n)}$.
   - Compute $m'_{int} \equiv c_2^{s'} \pmod{n}$
   - Compute $m' \equiv (m'_{int})^{u^t} \pmod{n}$
   - Check $\bar{m}' \stackrel{?}{=} lsb(m', k)$
   - If above step is correctly verified, return m', else
   - Check $\bar{m}' \stackrel{?}{=} lsb(-m', k)$
   - If above step is correctly verified, return $-m'$, else
   - return $\perp$.

5.4 Correctness of the Scheme

We provide a sketch of the correctness of the scheme.

Lemma 2. *Let n be the product of two prime numbers p and q where $p \equiv q \equiv 3 \pmod{4}$. Let $u = \frac{\phi(n)+4}{8}$. Then u is a positive integer.*

Proof. $p \equiv q \equiv 3 \pmod{4}$ implies $p = 4k_1 + 3$ and $q = 4k_2 + 3$ for some positive integers k_1 and k_2. So,

$$\frac{\phi(n)+4}{8} = \frac{pq - p - q + 1 + 4}{8} = 2k_1k_2 + k_1 + k_2 + 1$$

Hence, the lemma. □

Proposition 1. *Let n be the product of two prime numbers p and q where $p \equiv q \equiv 3 \pmod{4}$. Let $u = \frac{\phi(n)+4}{8}$. If $y \equiv x^2 \pmod{n}$ for some $x \in J(n)$, then $y^d \equiv x \pmod{n}$ or $y^d \equiv -x \pmod{n}$.*

Proof. We have

$$y^d \equiv (x^2)^{\frac{\phi(n)+4}{8}} = x^{\frac{\phi(n)+4}{4}} = x^{\frac{\phi(n)}{4}} x \pmod{n}.$$

If $x \in QR(n)$, $x^{\frac{\phi(n)}{4}} \equiv 1 \pmod{n}$. Hence, $y^d \equiv x^{\frac{\phi(n)}{4}}.x \equiv x \pmod{n}$.
If $x \in J(n) \setminus QR(n)$, $-x \in QR(n)$. So, $(-x)^{\frac{\phi(n)}{4}} \equiv 1 \pmod{n}$. Hence, $y^d \equiv (-x)^{\frac{\phi(n)}{4}}.(-x) \equiv (-x) \pmod{n}$. Hence, the proposition. □

Corollary 1. *Let n be the product of two prime numbers p and q where $p \equiv q \equiv 3 \pmod{4}$. Let $u = \frac{\phi(n)+4}{8}$. If $y \equiv x^{2^t} \pmod{n}$ for some $x \in J(n)$ and for some $t \geq 1$, then $y^{d^t} \equiv x \pmod{n}$ or $y^{d^t} \equiv -x \pmod{n}$.*

From Lemma 2 and Corollary 1, it is trivial to check the correctness of the scheme.

5.5 Security of the Encryption Algorithm $\mathcal{E}$

In this section, we will discuss the security of the encryption algorithm $\mathcal{E}$. To prove that $\mathcal{E}$ is IND-CPA secure, we propose two hardness assumptions.

1. **D-VRSA (Decisional-Variant of RSA):** Let n be the product of two odd prime numbers p and q, e be a number such that $\gcd(e, \phi(n)) = 1$, $\gamma \xleftarrow{\mathcal{R}} J(n) \setminus \{-1, 1\}$, $r \xleftarrow{\mathcal{R}} \mathbb{Z}_n$. Let $\bar{r} = msb(r, l-k)$ We say strong-D-VRSA problem to be hard if there exists no distinguisher $\mathcal{D}$ which can distinguish the distribution of two triples $(m, r^e \pmod n, m^{\bar{r}} \pmod n)$ from $(m, r^e \pmod n, \gamma^{\bar{r}} \pmod n)$ with non-negligible advantage, where $m \in J(n) \setminus \{-1, 1\}$ has been chosen by distinguisher $\mathcal{D}$. Formally,

$$\Big|\Pr[\gamma \xleftarrow{\mathcal{R}} J(n) \setminus \{-1,1\}, r \xleftarrow{\mathcal{R}} \mathbb{Z}_n; \mathcal{D}\left(m, r^e \pmod n, \gamma^{\bar{r}} \pmod n\right) = 1] - \Pr[r \xleftarrow{\mathcal{R}} \mathbb{Z}_n; \mathcal{D}\left(m, r^e \pmod n, m^{\bar{r}} \pmod n\right) = 1]\Big| < \epsilon$$

   where ϵ is a negligible quantity. Here probability is taken over random choices of γ, r and random coins of $\mathcal{D}$. We assume that D-VRSA is computationally hard.
2. **Mod-Bit-VRSA (Modified Bit-Variant of RSA):** Let n be the product of two odd prime numbers p and q, e be a number such that $\gcd(e, \phi(n)) = 1$, $z \xleftarrow{\mathcal{R}} \{0,1\}^k$, $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$. We say Mod-Bit-VRSA problem to be hard if there exists no distinguisher $\mathcal{D}$ which can distinguish the distribution of two triples $(m, (\bar{r}||z)^e \pmod n)$ from $(m, (\bar{r}||lsb(m,k))^e \pmod n)$ with non-negligible advantage, where $m \in \mathbb{Z}_n^*$ has been chosen by distinguisher $\mathcal{D}$. Formally,

$$\Big|\Pr\left[z \xleftarrow{\mathcal{R}} \{0,1\}^k, \bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}; \mathcal{D}\left(m, (\bar{r}||z)^e \pmod n\right) = 1\right] - \Pr\left[\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}; \mathcal{D}\left(m, (\bar{r}||lsb(m,k))^e \pmod n\right) = 1\right]\Big| < \epsilon$$

   where ϵ is a negligible quantity. Here probability is taken over random choices of $z, \bar{r}$ and random coins of $\mathcal{D}$. We assume that Mod-Bit-VRSA is computationally hard.

Now we will present the following result.

Theorem 2. *Encryption algorithm $\mathcal{E}$ is IND-CPA secure under D-VRSA and Mod-Bit-RSA assumption.*

Proof. We use sequence of games to prove that $\mathcal{E}$ is IND-CPA.
Game 0: [Real Game] Let there be an adversary which we model as deterministic algorithm that takes random coins as input sampled uniformly from the set R. We define Game 0 as a real game defined in IND-CPA security. In this game, adversary $\mathcal{A}$ chooses two equal length message m_0 and m_1 and gives them to challenger. Challenger then chooses a random bit $b \in \{0,1\}$ and encrypts the message m_b and gives the resulting ciphertext to $\mathcal{A}$. $\mathcal{A}$ then outputs the bit $\hat{b}$. We define Game 0 algorithmically as follows:

$x \xleftarrow{\mathcal{R}} R$, $(m_0, m_1, st) \leftarrow \mathcal{A}(x, n, e)$
$b \xleftarrow{\mathcal{R}} \{0,1\}$, $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$, $r \leftarrow \bar{r}||lsb(m_b, k)$, $c_1 \leftarrow r^e$, $c_2 \leftarrow m_b^{\bar{r}}$
$\hat{b} \leftarrow \mathcal{A}(x, n, e, c_1, c_2, st)$

Let S_0 be the event that $\mathcal{A}$ wins the Game 0, i.e., $b = \hat{b}$. Advantage of $\mathcal{A}$ is defined as

$$\mathcal{A}dv(\mathcal{A}) = \left|\Pr[S_0] - \frac{1}{2}\right| \tag{2}$$

Now, challenger will make a change in Game 0. Let the modified game be Game 1.

Game 1: [Transition based on the indistinguishability]. Here instead of taking $r = \bar{r}||lsb(m_b, k)$, challenger chooses a random string uniformly sampled from the set $\{0,1\}^k$ and then takes $r = \bar{r}||z$. Game 1 is defined algorithmically as follows:

$x \xleftarrow{\mathcal{R}} R$, $(m_0, m_1, st) \leftarrow \mathcal{A}(x, n, e)$
$b \xleftarrow{\mathcal{R}} \{0,1\}$, $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$, $z \xleftarrow{\mathcal{R}} \{0,1\}^k$, $r \leftarrow \bar{r}||z$, $c_1 \leftarrow r^e$, $c_2 \leftarrow m_b^{\bar{r}}$
$\hat{b} \leftarrow \mathcal{A}(x, n, e, c_1, c_2, st)$

Let S_1 be the event that $b = \hat{b}$. It is clear from Game 0 and Game 1 that

$$|\Pr[S_0] - \Pr[S_1]| < \epsilon_1 \tag{3}$$

where ϵ_1 is the advantage in Mod-Bit-VRSA problem. Now, Challenger will make a change in Game 1. Let the modified game be Game 2.

Game 2: [Transition based upon the indistinguishability]. In this game, instead of taking $m_b^{\bar{r}}$, challenger chooses a random number from the set $J(n) \setminus \{-1, 1\}$, say γ, and takes $\gamma^{\bar{r}}$. Game 2 is defined as follows:

$x \xleftarrow{\mathcal{R}} R$, $(m_0, m_1, st) \leftarrow \mathcal{A}(x, n, e)$
$b \xleftarrow{\mathcal{R}} \{0,1\}$, $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$, $z \xleftarrow{\mathcal{R}} \{0,1\}^k$, $\gamma \xleftarrow{\mathcal{R}} J(n) \setminus \{-1, 1\}$,
$r \leftarrow \bar{r}||z$, $c_1 \leftarrow r^e$, $c_2 \leftarrow \gamma^{\bar{r}}$
$\hat{b} \leftarrow \mathcal{A}(x, n, e, c_1, c_2, st)$

Let S_2 be the event that $b = \hat{b}$. It is clear from Game 1 and Game 2 that

$$|\Pr[S_1] - \Pr[S_2]| < \epsilon_2. \tag{4}$$

where ϵ_2 is the advantage in D-VRSA problem.
In Game 2, x, n, e, c_1 and c_2 all are independent of the bit b, hence adversary's output $\hat{b}$ is independent of the hidden bit b. Hence, $\Pr[S_2] = 1/2$.
So, from equation (2),(3) and (4) we get,

$$\begin{aligned}\left|\Pr[S_0] - \frac{1}{2}\right| &= |\Pr[S_0] - \Pr[S_1] + \Pr[S_1] - \Pr[S_2]| \\ &\leq |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_2]| \\ &< \epsilon_1 + \epsilon_2 \end{aligned} \tag{5}$$

So, advantage of adversary $\mathcal{A}$ be

$$\mathcal{A}dv(\mathcal{A}) = |\Pr[S_0] - 1/2| < \epsilon_1 + \epsilon_2.$$

Hence the result. □

Now we will prove that $\mathcal{E}$ is not IND-CCVA secure.

Lemma 3. *$\mathcal{E}$ is not IND-CCVA secure.*

$$Exp_{\mathcal{A},\mathcal{C}}^{\mathcal{E},IND-CCVA}(n,e).$$

1 $x \xleftarrow{\mathcal{R}} R$;
2 $(m_0, m_1, st) \leftarrow \mathcal{A}(x, n, e)$ such that $lsb(m_0, k) = 0^k$ and $lsb(m_1, k) = 01||\{0,1\}^{k-3}0$;
3 $b \xleftarrow{\mathcal{R}} \{0,1\}$, $\bar{r} \xleftarrow{\mathcal{R}} \{0,1\}^{l-k}$, $z = lsb(m_b, k)$;
4 $C_b \equiv (c_{b1}, c_{b2}) \leftarrow Enc(m_b, PK)$, where $c_{b1} = (\bar{r}||z)^e$ and $c_{b2} = m_b^{\bar{r}}$;
5 $C_b \leftarrow \mathcal{C}(n, e, m_b)$;
6 $C_b' \equiv (2^e.c_{b1}, c_{b2}^2) \leftarrow \mathcal{A}(x, n, e, C_b, st)$;
7 If Decision Oracle outputs 1, return m_0 ;
8 Else output m_1;

Proof. Let $Exp_{\mathcal{A},\mathcal{C}}^{\mathcal{E},IND-CCVA}$ denotes an experiment where PPT adversary $\mathcal{A}$ breaks the IND-CCVA security of $\mathcal{E}$ in the game played with challenger $\mathcal{C}$. We have named this experiment as $Exp_{\mathcal{A},\mathcal{C}}^{\mathcal{E},IND-CCVA}$.

Analysis

Now, we analyse the success probability of adversary $\mathcal{A}$ in the experiment $Exp_{\mathcal{A},\mathcal{C}}^{\mathcal{E},IND-CCVA}(n,e)$. There will be total 4 cases, depending on the most significant bit of $\bar{r}$ and the message. The ciphertext generated by adversary $\mathcal{A}$ will be a valid ciphertext if most significant bit of $\bar{r}$ is 0 and message is m_0 and for rest of the cases, the ciphertext will be invalid.
Case 1. If $m_b = m_0$ and the most significant bit of $\bar{r}$ be 0, then
$\bar{r}||z = 0\bar{r}_1||0^k$, where $\bar{r}_1 \in \{0,1\}^{l-k-1}$. So,

$$2(\bar{r}||z) = \bar{r}_1 0||0^k = 2\bar{r}||z$$

Since $\bar{r}||z < n/2$, hence

$$2^e(\bar{r}||z)^e \pmod n \equiv (2(\bar{r}||z))^e \pmod n \equiv (2\bar{r}||z)^e \pmod n$$

Since $C_0' \equiv (2^e c_{01}, c_{02}^2) \equiv (2^e(\bar{r}||z)^e, m_0^{2\bar{r}}) \equiv ((2\bar{r}||z)^e, m_0^{2\bar{r}})$ and $lsb(m_0, k) = z = 0^k$, Ciphertext Verification Oracle will return valid 1. Adversary $\mathcal{A}$ will output m_0 as per experiment which is a correct guess with probability 1.
Case 2. If $m_b = m_1$ and the most significant bit of r is 0, then
$\bar{r}||z = 0\bar{r}_1||01\bar{m}_1$, where $\bar{r}_1 \in \{0,1\}^{l-k-1}$ and $01\bar{m}_1 = lsb(m_1, k)$. So,

$$2(\bar{r}||z) = \bar{r}0||1\bar{m}_1 0 = 2\bar{r}||z' \tag{6}$$

Note that $z \neq z'$ as $msb(z) = 0$ but $msb(z') = 1$.
Since $\bar{r}||z < n/2$, hence

$$2^e(\bar{r}||z)^e \pmod n \equiv (2(\bar{r}||z))^e \pmod n \equiv (2\bar{r}||z')^e \pmod n$$

Here, $C_1' \equiv (2^e c_{11}, c_{12}^2) \equiv (2^e(\bar{r}||z)^e, m_1^{2\bar{r}}) \equiv ((2\bar{r}||z')^e, m_1^{2\bar{r}})$. In decryption algorithm, during validity checking, candidate messages m_1 and $-m_1$ both will have $lsb(m_1, k) = z \neq z'$ as $msb(z) \neq msb(z')$ and $lsb(-m_1, k) \neq z'$ as $lsb(-m_1) = 1 \neq 0 = lsb(z')$. Hence, Ciphertext Verification Oracle will return 0. Adversary $\mathcal{A}$ will output m_1 as per experiment which is a correct guess with probability 1.
Case 3. If $m_b = m_0$ and the most significant bit of r is 1, then $\bar{r}||z > n/2$. It is easy to check that

$$2^e(\bar{r}||z)^e \pmod n \equiv (2(\bar{r}||z))^e \pmod n \equiv (\bar{r}'||\bar{z})^e \pmod n$$

where $|\bar{z}| = |z| = k$, $z \neq \bar{z}$, $|\bar{r}'| = |\bar{r}|$ and $\bar{r}' \neq r$. Note that $C_0' \equiv \left(2^e c_{01}, c_{02}^2\right) \equiv \left(2^e(\bar{r}||z)^e, m_0^{2\bar{r}}\right) \equiv \left((\bar{r}'||\bar{z})^e, m_0^{2\bar{r}}\right)$. In decryption algorithm, during validity checking, candidate messages m' and $-m'$

will have $lsb(m', k) = \bar{z}$ or $lsb(-m', k) = \bar{z}$ with probability p, say. Therefore, Ciphertext Verification Oracle will return 1 with probability p. In this case, if adversary $\mathcal{A}$ guesses the message m_0, then it will be a correct guess with probability p.
Case 4. If $m_b = m_1$ and the most significant bit of r is 1, then $\bar{r}||z > n/2$. It is easy to check that

$$2^e(\bar{r}||z)^e \pmod{n} \equiv (2(\bar{r}||z))^e \pmod{n} \equiv (\bar{r}'||\bar{z})^e \pmod{n}$$

where $|\bar{z}| = |z| = k$, $z \neq \bar{z}$, $|\bar{r}'| = |\bar{r}|$ and $\bar{r}' \neq r$. Clearly $C_0' \equiv \left(2^e c_{01}, c_{02}^2\right) \equiv \left(2^e(\bar{r}||z)^e, m_0^{2\bar{r}}\right) \equiv \left((\bar{r}'||\bar{z})^e, m_0^{2\bar{r}}\right)$. In decryption algorithm, during validity checking, candidate messages m' and $-m'$ will have $lsb(m', k) = \bar{z}$ or $lsb(-m', k) = \bar{z}$ with probability p. Therefore, Ciphertext Verification Oracle will return 0 with probability close to $1 - p$. In this case, if adversary $\mathcal{A}$ guesses the message m_1, then it will be a correct guess with probability close to $1 - p$.
Adversary will output correct answer with probability

$$\frac{1}{4} \times 1 + \frac{1}{4} \times 1 + \frac{1}{4} \times p + \frac{1}{4} \times (1 - p) = \frac{3}{4}$$

Hence, encryption algorithm $\mathcal{E}$ is not IND-CCVA secure. □

References

1. W. Alexi, B. Chor, O. Goldreich, and P. Schnorr. Bit security of RSA and Rabin functions. SIAM Journal of Computing, 17(2):194-209, Apr. 1988.
2. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In Proc. Eurocrypt'94, pages 92–111, 1994.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In Proc. Crypto'98, pages 26–45, 1998.
4. D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In Proc. Crypto'98, pages 1–12, 1998.
5. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Proc. Crypto'98, pages 13–25, 1998.
6. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. SIAM J. Computing 30(2): 391-437 (2000).
7. A. O. Freier, P. Karlton and P. C. Kocher. The SSL Protocol. Version 3.0.
8. O. Goldreich. A Uniform Complexity Treatment of Encryption and Zero-Knowledge. J. Cryptology 6(1): 21-35 (1993).
9. O. Goldreich. Foundations of Cryptography, vol. 2. Basic Applications. Cambridge University Press, 2004.
10. S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network. In Proc. 23rd IEEE Symp. on Foundations of Comp. Science, pages 134-144, Chicago, 1982.
11. S. Goldwasser and S. Micalli. Probabilistic encryption. Journal of Computer and System Sciences. 28(2): 270–299, 1984.
12. J. Håstad and M. Näslund The security of individual RSA bits. manuscript, 1998.
13. J. Katz and Y. Lindell. Introduction to Modern Cryptography. Chapman & Hall/CRC, 2007.
14. S. Micali, C. Rackoff, and B. Sloan. The Notion of Security for Probabilistic Cryptosystems. SIAM J. Computing 17(2): 412-426 (1988).
15. M. Naor and M. Yung. Public-key cryptosystem provably secure against chosen ciphertext attacks. In Proc. STOC, pages 427–437, 1990.
16. P. Pallier. Public-Key cryptosystem based on composite degree residuosity classes. In Proc. Eurocrypt'99, pages 223–238, 1999.
17. Public-Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard. RSA Security Inc., 2002.
18. M. Rabin. Digitalized signatures as intractable as factorization. Technical Report TR-212, MIT/LCS, 1979.

19. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Proc. Crypto'91, pages 433–444, 1992.
20. R. L. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of ACM, 21(2):158–164, February 1978.
21. RSA Data Security, Inc. PKCS #1: RSA Encryption Standard. Redwood City, CA, Nov. 1993. Version 1.5.
22. V. Shoup. Why chosen ciphertext security matters. Technical Report RZ 3076, IBM Zurich, 1998.
23. V. Shoup. Sequences of games: a tool for taming complexity in security proofs, Cryptology ePrint Archive: Report 2004/332, 2004.