National University of Singapore
CS4230/CS5430: Foundations of Modern Cryptography
AY 2022/23, Sem 2

Final Project
Student 1: Wong Choon Wei
Student 2: Nguyen Doan Phuong Anh

Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks

Moni Naor & Moti Yung

We have covered a range of public key encryption key in class like the Merkle Protocol, to the Diffie-Hellman/ElGamal Encryption which takes advantage of algebraic properties to assure security, the Rivest–Shamir–Adleman (RSA) and Rabin Trapdoor Permutations which builds security from number theory and finally the Public-Key Encryption scheme based on Learning with Errors (LWE). However, while these schemes are known to provide strong security in many scenarios, we have not explored whether these encryption schemes are provably secure against chosen ciphertext attacks (CCA). In this paper, we address this issue by examining public-key cryptosystems and provide a rigorous proof of their security against CCA. This paper provides a self-contained continuation to the original article by Naor & Yung who were the first to provide a formal definition and proof of security for public-key cryptosystems against CCA.

1 Introduction

The development of public-key cryptography by Diffie and Hellman in 1976 was a major breakthrough in the field of cryptography, as it allowed for secure communication without the need for a shared secret key. However, the security of public-key cryptosystems can be compromised by an adversary who has access to both the ciphertext and the encryption key. One of the most powerful attacks against public-key cryptosystems is the chosen ciphertext attack (CCA), where the adversary can submit ciphertexts of its own choosing to be decrypted and observe the resulting plaintexts. This type of attack can be devastating, as it can allow the adversary to obtain sensitive information about the plaintext.

Previously, in order to construct message transmission systems secure against chosen ciphertext attacks, the public-key model was relaxed, and some interaction was assumed between the users prior to the actual transmission of the message. However, this is not ideal for scenarios where interaction is limited or not possible.

To address this challenge, Naor and Yung proposed a new construction for a public-key cryptosystem that is provably secure against CCA, assuming the existence of a public-key cryptosystem that is secure against passive eavesdropping and a non-interactive zero-knowledge proof system in the shared string model. Their work provides a solution to one of the most important challenges in public-key cryptography and opens up new possibilities for secure communication over insecure channels.

The motivation for their work came from the potential of applying non-interactive proof systems to achieve chosen ciphertext security, as pointed out by Blum, Feldman, and Micali in their introduction of single-theorem non-interactive zero-knowledge proof systems for language membership [SMP87]. Some related work explored the idea, but Silvio Micali interpreted the claim to mean that the scheme requires interaction between users, whereas Naor and Yung's scheme is a "one and a half pass" scheme that requires minimal interaction.

The security of electronic communication is crucial in today's interconnected world. The ability to communicate securely over insecure channels without needing to share a secret key beforehand has been a game-changer for electronic communication. However, the security of public-key cryptosystems is under constant threat from various attacks. CCA is one of the most powerful attacks against public-key cryptosystems, and until Naor and Yung's work, no provably secure solution had been proposed before.

In this exposition paper, we will cover Naor and Yung's paper "Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attcks", which proposes a new construction for a public-key cryptosystem that is provably secure against CCA. We will begin by providing a brief overview of public-key cryptosystems and their security against various types of attacks. We will then dive into Naor and Yung's construction and provide a technical overview of their approach, identifying the primary technical challenges and how they are ovecrome. Finally, we will discuss the primary questions left open by the paper and explore some potential approaches that could answer them.

2 Technical background

We repeat the definition of public-key encryption schemes, CCA, and introduce the notion of probabilistic encryption schemes, non-interactive zero-knowledge proof systems.

2.1 Public-key encryption schemes

Definition 2.1 (Public-key encryption schemes)**.** A *public-key encryption* scheme consists of three algorithms:

- $\mathsf{KeyGen}(1^\lambda)$: outputs a public key and a secret key (pk, sk)
- $\mathsf{Enc}(1^\lambda, pk, m)$: outputs ciphertext c
- $\mathsf{Dec}(1^\lambda, sk, c)$: outputs message $\hat{m}$

2.2 Chosen Ciphertext Attacks - CCA

IND – CCA(λ) :

1. Challenger samples a random bit $b \leftarrow \{0, 1\}$, and invokes $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Challenger sends the public key pk to Adversary.
3. For $i \in \mathsf{poly}(\lambda)$:
   - Adversary sends some ciphertext c_i to the Challenger.
   - Challenger responds with the corresponding decryption $\mathsf{Dec}(1^\lambda, sk, c_i)$ to the Adversary.
4. Adversary sends to messages m_0, m_1 to the Challenger.
5. Challenger sends the challenger ciphertext $c = \mathsf{Enc}(1^\lambda, pk, m_b)$.
6. Adversary outputs a guess bit b'.
7. Adversary wins iff $b' = b$.

Definition 2.2 (CCA security)**.** A Public-key encryption scheme is CCA-secure if, for any PPT adversary A, there exists a negligible function ν such that for all λ

$$Pr[A \text{ wins IND-CCA}(\lambda)] < \frac{1}{2} + \nu(v)$$

2.3 Probabilistic encryption schemes

Probabilistic encryption schemes are analogous to CPA-secure PKE scheme. In particular, the input bit b for Enc in the definition below plays the same role as the secret random bit, which is sampled by the challenger in the IND-CPA game.

Definition 2.3 (Probabilistic encryption schemes)**.** A *probabilistic encryption* scheme consists of three polynomial time algorithms:

- $\mathsf{KeyGen}(1^\lambda)$: a probabilistic algorithm that outputs a public key and a secret key (pk, sk)

- $\mathsf{Enc}(1^\lambda, pk, b, r)$: a probabilistic algorithm that outputs ciphertext c, where $b \in \{0,1\}$ and $r \in \{0,1\}^{\mathsf{poly}(\lambda)}$
- $\mathsf{Dec}(1^\lambda, sk, c)$: outputs $\hat{b} \in \{0,1\}$

We want *probabilistic encryption schemes* to have two properties:

- **Correctness:** $\forall \lambda\,, \forall b \in \{0,1\}\,, \forall r \in \{0,1\}^{\mathsf{poly}(\lambda)}$

$$Pr_{(pk,sk)\leftarrow \mathsf{KeyGen}(1^\lambda)}[\mathsf{Dec}(1^\lambda, sk, \mathsf{Enc}(1^\lambda, pk, b, r)) = b] = 1$$

- **Indistinguishability:** For any PPT Algorithm A, there exists a negligible function ν such that for all λ with $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $r \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}$:

$$Pr[A(1^\lambda, pk, \mathsf{Enc}(1^\lambda, pk, 0, r))] - Pr[A(1^\lambda, pk, \mathsf{Enc}(1^\lambda, pk, 1, r))] < \nu(\lambda)$$

Conjecture 1. *There exists a probabilistic encryption scheme that satisfies both correctness and indistinguishability.*

From the Goldreich-Levin Hardcore Predicate, it follows that if trapdoor permutations exists, then such schemes exist.

2.4 Non-interactive zero-knowledge proof systems

In *non-interactive zero-knowledge proof systems*, we first run the protocol with a simulator (who is just a verifier, but the random choices are done differently), and then we can give the transcript of the protocol to anyone and convince them that the proof is real. The key benefit of non-interactive zero-knowledge proofs is that they can be used in situations where there is no possibility of interaction between the prover and verifier.

Notation:

- For a distribution $\mathcal{Q}$, let $x \in_R \mathcal{Q}$ denote that x is generated by distribution $\mathcal{Q}$.
- An *ensemble* of probability distributions $\mathcal{Q}(x)$ is polynomial time sampleable if there is a probabilistic polynomial (in $|x|$) time machine that on input x its output is distributed according to $\mathcal{Q}(x)$.

Definition 2.4 (Computationally indistinguishable distributions)**.** Two ensembles of probability distributions $\mathcal{A}(x)$ and $\mathcal{B}(x)$ are polynomial time *indistinguishable* if for any probabilistic polynomial time machine $\mathcal{C}$, called the *distinguisher*, that acts as follows: first $x \in_R C(\lambda)$ is generated and then $\mathcal{C}$ is given the output generated by either $\mathcal{A}(x)$ or $\mathcal{B}(x)$. Then, there exists a negligible function ν such that for all λ

$$|Pr[\mathcal{C}(x,y) = 1 | y \in_R \mathcal{A}(x)] - Pr[\mathcal{C}(x,y) = 1 | y \in_R \mathcal{B}(x)]| < \nu(\lambda)$$

Overview: A (single theorem) non-interactive proof system for a language L allows a prover $\mathcal{P}$ to prove membership in L to a verifier $\mathcal{V}$ for any $x \in L$. $\mathcal{P}$ and $\mathcal{V}$ initially share a string R of length polynomial in the security parameter λ. To prove membership of a string x in $L \cap \{0,1\}^\lambda$, $\mathcal{P}$ sends a message p as a proof of membership. $\mathcal{V}$ decides if to accept or to reject the proof. The shared string R is generated according to some distribution $\mathcal{U}(\lambda)$ that can be generated by a probabilistic polynomial time machine.

Remark 2.1. *For the scheme described in Section 3.3 to work, the distribution $\mathcal{U}$ needs to be uniform and this uniformity condition is adopted from [Gol93].*

Notation:

- An *overwhelming probability* is a probability, which is at least $1 - \frac{1}{\mathsf{poly}(\lambda)}$ for infinitely many λ's.
- Let L be in NP. For any $x \in L$ there is a set of *witnesses* for its membership denoted as $WL(x) = \{z | z \text{ is a witness for } x\}$. For the proof system to be of any use, $\mathcal{P}$ should be able to operate in polynomial time if it is given a witness $z \in WL(x)$. Note that z is not available to $\mathcal{V}$.

- Let $\mathcal{P}(x, z, R)$ denote the distribution of the proofs that $\mathcal{P}$ generates on input x, witness z and shared string R.
- For any proof $p \in \mathcal{P}(x, z, R)$, we called (R, p) a *conversation.*
- For any $x \in L$ and $z \in WL(x)$, let $\mathcal{CONV}(x, z)$ denote the probability distribution of conversations (R, p) where $R \in \mathcal{U}(\lambda)$ is a (random) shared string and $p \in \mathcal{P}(x, z, R)$ is a proof.
- Let $\mathcal{S}(x)$ be the distribution of the conversation (R, p) that a simulator $\mathcal{S}$ generates on input x, and let $\mathcal{S}_R(x)$ be the distribution of the R part of the conversation.
- Denote $ACCEPT(R, x) = \{p | \mathcal{V}$ accepts on input $R, x, p\}$.
- Denote $REJECT(R, x) = \{p | \mathcal{V}$ rejects on input $R, x, p\}$.

Definition 2.5 (Non-interactive zero-knowledge proof system)**.** A *(single theorem) non-interactive zero-knowledge proof system* for the language $L \in NP$ consists of 3 components: a probabilistic machine $\mathcal{P}$, a polynomial time machine $\mathcal{V}$ and a polynomial time sampleable probability distribution $\mathcal{U}$ that satisfies the following properties:

- **Completeness:** (If $x \in L$ then $\mathcal{P}$ generates a proof that $\mathcal{V}$ accepts).
  For all $x \in L \cap \{0,1\}^\lambda$, for all $z \in WL(x)$, with overwhelming probability for $R \in_R \mathcal{U}(\lambda)$ and $p \in_R \mathcal{P}(x, z, R)$, p is in $ACCEPT(R, x)$. *Note that randomness is over the choices of the shared string R and the inner coin flips of $\mathcal{P}$.*
- **Soundness:** (If $y \notin L$ then no prover can generate a proof that $\mathcal{V}$ accepts).
  For all $y \notin L \cap \{0,1\}^\lambda$, with overwhelming probability over $R \in_R \mathcal{U}(\lambda)$ and for all $p \in \{0,1\}^*$, p is in $REJECT(R, x)$. *Note that randomness is over the choices of the shared string R.*
- **Zero-knowledge:** (There is a probabilistic polynomial time simulator $\mathcal{S}$ for the system).
  For all probabilistic polynomial time algorithm $\mathcal{C}$, if $\mathcal{C}$ generates $x \in L$ and $z \in WL(x)$ then there exists a negligible function ν such that for all λ

$$|Pr[\mathcal{C}(w) = 1 | w \in_R \mathcal{S}(x)] - Pr[\mathcal{C}(w) = 1 | w \in_R \mathcal{CONV}(x, z)]| < \nu(\lambda)$$

Conjecture 2. *For all $L \in NP$, there exists a non-interactive zero-knowledge proof system for L as defined above.*

This assumption is known to be true assuming the intractability of quadratic residuosity [SMP87] or given any trapdoor one-way permutation [FLS99a].

3 The scheme

The scheme consists of two main parts: two keys (double encryption) and a random string generated by some distribution. Each message is encrypted according to each encryption key and an appended proof. For decryption, it is first verified in the non-interactive proof system that the two encryptions are consistent, and only then is the message decrypted. However, the non-interactive proof systems as defined above are not strong enough.

3.1 Strengthening non-interactive zero-knowledge proof systems

It is known that if a PKE scheme is CPA-secure, then it is also Multi-CPA-secure. Using the similar idea, we want to maintain the zero-knowledge property while running several proof systems concurrently. Consider the following "generic" transformation: Let

$$(\mathcal{P}_1, \mathcal{V}_1, \mathcal{U}_1), (\mathcal{P}_2, \mathcal{V}_2, \mathcal{U}_2), \ldots, (\mathcal{P}_n, \mathcal{V}_n, \mathcal{U}_n)$$

be non-interactive proof systems for a language L. We construct $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ as follows: $\mathcal{U}$ generates a string $R = R_1, R_2, .., R_n$ such that $R_i \in_R \mathcal{U}_i(\lambda)$ for all i. On input x and $z \in WL(x)$, prover $\mathcal{P}$ generates $p = (p_1, p_2, \ldots p_n)$ where $p_i \in_R \mathcal{P}_i(x, z, R_i)$ for all i. Verifier $\mathcal{V}$ runs $\mathcal{V}_i$ on x, R_i, p_i for all i. $\mathcal{V}$ accepts if all the $\mathcal{V}_i$'s accept.

Lemma 3.1. *$(\mathcal{P},\mathcal{V},\mathcal{U})$ is a non-interactive zero knowledge proof system for L.*

Proof: If a sequence of n conversations $\mathcal{CONV}_i(x,z)$ is distinguishable from a sequence of n simulators $\mathcal{S}_i(x)$ with probability at least ϵ then by using the following hybrid argument

$$(\mathcal{CONV}_1(x,z),\mathcal{CONV}_2(x,z))\,,\ldots\mathcal{CONV}_i(x,z),\mathcal{S}_{i+1}(x),\ldots\mathcal{S}_n(x)$$

then there exists j such that $\mathcal{CONV}_j(x,z)$ is distinguishable from $\mathcal{S}_j(x)$ with probability at least $\frac{\epsilon}{n}$.

Strong soundness

Since R is given as part of the public key, and x, which should be a consistent double encryption, is selected by the sender afterward. The soundness condition then should hold even if x is chosen after R is known. Hence, strong soundness requirement is: with overwhelming probability over $R \in_R \mathcal{U}(\lambda), \forall y \notin L_n, \forall p \in \{0,1\}^*$, $p \in REJECT(R,y)$. To tackle this issue, we use a quantifier swapping technique in [Zac86]: For words of length n, R would actually be a sequence of $2n$ strings $R_1, R_2, \ldots R_{2n}$, where each $R_i \in_R \mathcal{U}(\lambda)$. To prove that $x \in L_n$, for all $1 \leq i \leq 2n$ a proof p_i in $ACCEPT\,(R_i,x,z)$ is given. To verify a proof, invoke $\mathcal{V}$ on each proof p_i and accept if for all $1 \leq i \leq 2n$ $\mathcal{V}_i$ accepted. By Lemma 3.1 the resulting scheme is still a non-interactive zero-knowledge proof system.

Valid distributions

It follows from the strong soundness condition that if $R \in_R \mathcal{U}$, then for all $y \notin L, REJECT(R,y)$ contains all p 's with high probability. However, $\mathcal{S}_R(x)$, the distribution of the part R that the simulator generates on x is not necessary $\mathcal{U}$. Thus, it should also be impossible to find a $y \notin L$ and p such that $p \notin REJECT(R,y)$.

Definition 3.1 (Invalid word)**.** An *invalid word* for R in a proof system $\mathcal{P},\mathcal{V},\mathcal{U}$ is a pair (y,p) such that $y \notin L$ yet $p \in ACCEPT(R,y)$.

Using this definition, in a proof system with strong soundness with overwhelming probability there are no invalid words for $R \in_R \mathcal{U}$.

We show that if $x \in L$ then it should be difficult to find invalid words for $R \in_R S_r(x)$, and that for any x the invalid words of $R \in_r S_R(x)$ can be identified.

Definition 3.2. Let L be a language for which a non-interactive zero-knowledge proof system $(\mathcal{P},\mathcal{V},\mathcal{U})$ exists. A family of distributions $\{\mathcal{Q}_R(x) \mid x \in L\}$ such that $\mathcal{Q}_R(x)$ generates strings R of the length required by the proof system for words of length $|x|$ is valid if there is no probabilistic polynomial time machine $\mathcal{C}$ that: first produces $x \in L$, then on input $R \in_R \mathcal{Q}_R(x)$ can find with non-negligible probability a $y \notin L$ and a proof $p \notin REJECT(R,y)$. The probability of the success of $\mathcal{C}$ is taken over $\mathcal{Q}_R(x)$ and the inner coin flips of $\mathcal{C}$.

Let $(\mathcal{P},\mathcal{V},\mathcal{U})$ be a proof system satisfying strong soundness with valid distribution $\mathcal{U}$.

We construct a new system $(\mathcal{P}',\mathcal{V}',\mathcal{U}')$ such that the distribution $\mathcal{S}'_R$ is also valid. Firstly, to show that $x \in L, \mathcal{P}'$ picks a random subset $J \subset \{1\ldots n\}$ of size $\frac{n}{2}$. For each i a proof $p_i \in_R ACCEPT\,(R_i,x,z)$ is generated. The proof $p = (p_{i_1}, p_{i_2}, \ldots p_{i_{\frac{n}{2}}})$ where $i_j \in J$. To verify a proof, for each $i_j \in J$ run $\mathcal{V}$ on p_{i_j} with R_{i_j}. Accept if for all $1 \leq j \leq \frac{n}{2}\mathcal{V}$ accepted; $ACCEPT'(R,x)$ is defined accordingly.

The simulator $\mathcal{S}'$ invokes the simulator for the old system, $\mathcal{S}$, in $\frac{n}{2}$ times and generating $\frac{n}{2}$ strings by distribution $\mathcal{U}$. The outputs of S and the outputs of $\mathcal{U}$ are randomly shuffled to generate the simulated R. The subset J and the proof p are chosen appropriately. Similar to Lemma 3.1, the resulting system is still a non-interactive zero-knowledge proof system for L.

Claim 3.1. *If there exists a probabilistic polynomial time machine $\mathcal{M}$ that given $R = R_1, R_2, \ldots R_n$ can find the $\frac{n}{2}$ R_i's that were generated by $\mathcal{S}$ with non-negligible probability, then it can be used to distinguish between the output of the simulator $\mathcal{S}$ and true conversations.*

Proof: Assume for contradiction that $\mathcal{M}$ has some non-negligible probability δ of finding the outputs of $\mathcal{S}$. Suppose that instead of $\frac{n}{2}$ outputs of $\mathcal{S}$ randomly shuffled with $\frac{n}{2}$ outputs of $\mathcal{U}, \mathcal{M}$ is given two sets of $\frac{n}{2}$ strings, the first set containing i outputs of $\mathcal{S}_R(x)$ and $\frac{n}{2} - i$ outputs of $\mathcal{U}$ and the second

set contains only outputs of $\mathcal{U}$. The two sets are randomly shuffled. Let q_i be the probability that $\mathcal{M}$ succeeds in finding the correct partition to the two sets. Then, $q_0 = \frac{1}{\binom{n}{n/2}}$ and by assumption $q_{n/2} > \delta$. Hence, there exists some $1 \leq i < \frac{n}{2}$ such that $q_{i+1} - q_i > \frac{\delta}{n}$

Add a given string R to a set composed of i outputs of $\mathcal{S}$ and $\frac{n}{2} - i - 1$ outputs of $\mathcal{U}$ and randomly shuffle it with a set of $\frac{n}{2}$ outputs of $\mathcal{U}$ and give it to $\mathcal{M}$. If $\mathcal{M}$ guesses the right partition, then guess that R is not random. Otherwise flip a coin.

Claim 3.2. *The distribution S'_R is valid*

Proof: If there is a probabilistic polynomial-time machine $\mathcal{M}$ such that given the output of $\mathcal{S}', R = R_1, R_2, \ldots R_n$, then with non negligible probability $\mathcal{M}$ can find an invalid word (y, p). Given that $\mathcal{M}$ found such a y and p, then with overwhelming probability the set of indices J that is used in p is exactly those indices i for which R_i was an output of $\mathcal{S}$, since for the random R_i 's generated by $\mathcal{U}$, with overwhelming probability there are no $p_i \notin REJECT(R_i, y)$ (by the strong soundness requirement). Hence, we can use $\mathcal{M}$ to separate the outputs of S from the random ones and by the previous claim to distinguish between $\mathcal{S}(x)$ and $\mathcal{CONV}(x, z)$.

We now show that in case $x \notin L$ there is a method for recognizing the invalid words.

Claim 3.3. *There is a procedure M that can be run by a machine having as input $\mathcal{S}'$'s random tape such that for any probabilistic polynomial time machine $\mathcal{C}$ that first creates x, then gets $R \in_R S'_R(x)$ and then produces a word $w = (y, p)$ such that $p \in ACCEPT'(R, y)$ the following properties hold:*

- *if $\mathcal{C}$ has a non-negligible probability to output $x \notin L$ and then given $R \in_R S'_R(x)$ $\mathcal{C}$ finds an invalid word w for R, then M recognizes an invalid w as such with overwhelming probability.*
- *if $\mathcal{C}$ outputs $x \in L$ with non-negligible probability, then with overwhelming probability M does not recognize falsely as invalid a word that $\mathcal{C}$ outputs given $R \in_R S'_R(x)$ for $x \in L$*

Proof: The method to recognize a word (y, p) as invalid for $R = R_1, R_2, \ldots R_n$ is to check if p uses exactly those indices that were generated by $\mathcal{S}$. (This can be done efficiently by a machine M that has $\mathcal{S}'$'s random tape, since it can simulate $\mathcal{S}'$ directly using $\mathcal{S}$ as a black box and it has access to the choice of J on the random tape). Since $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ satisfies strong soundness, an invalid word for $R = R_1, R_2, \ldots, R_n$ uses with overwhelming probability those R_i 's that were generated by $\mathcal{S}$. Hence, strong soundness still holds. By the previous claim ($\mathcal{S}'_R$ is valid) it is obvious that if $x \in L$, with overwhelming probability $\mathcal{C}$ will not output a word that is recognized as invalid by this method.

Theorem 3.2 (Certifying system)**.** *Any non-interactive zero-knowledge proof system for a language L as in Definition 2.5 can be converted to one with the following properties.*

- ***Strong soundness:*** *$\forall y \notin L$, with overwhelming probability over $R \in_R \mathcal{U}(\lambda)$ and $\forall p \in \{0,1\}^*$, p is in $REJECT(R, y)$.*
- ***Validity:*** *the family of distributions $\{S'_R(x) | x \in L\}$ is valid.*
- ***Recognizability:*** *there is an efficient method for recognizing invalid words.*

We call a proof system satisfying those three properties a certifying system.

3.2 Consistent double encryption

Definition 3.3 (Consistent double encryption)**.** Let $(\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ be the probabilistic encryption schemes as in Definition 2.3. For public keys e_1, e_2, a *consistent double encryption* is

$$w = \mathsf{Enc}'(1^\lambda, pk_1, b, r_1), \mathsf{Enc}'(1^\lambda, pk_2, b, r_2)$$

for some $b \in \{0,1\}$ and $r_1, r_2 \in \{0,1\}^{\mathsf{poly}(\lambda)}$.

3.3 The scheme construction

The language of consistent double encryptions

$$L = \{e_1, e_2, w | \text{ is a consistent double encryption}\}$$

is in NP. For a given word $w = \mathsf{Enc}'(1^\lambda, pk_1, b, r_1), \mathsf{Enc}'(1^\lambda, pk_2, b, r_2)$ the pair r_1, r_2 is a witness for its membership in L. By ..., there exists a certifying system $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ for L.

We will use this certifying system $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ for L together with the probabilistic encryption schemes $(\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ defined in Definition 2.3 to construct the public-key encryption scheme as follows:

- $\mathsf{KeyGen}(1^\lambda)$:
  - invokes $\mathsf{KeyGen}'(1^\lambda)$ twice to generate (pk_1, sk_1) and (pk_2, sk_2)
  - samples $R \in_R \mathcal{U}(1^\lambda)$
  - outputs (pk, sk) where $pk = \langle pk_1, pk_2, R\rangle$ and $sk = \langle sk_1, sk_1\rangle$
- $\mathsf{Enc}(1^\lambda, pk, m)$: let $m = b_1 b_2 ... b_k$ where $b_i \in \{0, 1\}$
  - for $i \in [k]$:
    1. samples $r_{i_1}, r_{i_2} \in_R \{0,1\}^{\mathsf{poly}(\lambda)}$
    2. computes $c_i = c_{i_1}, c_{i_2}$ where $c_{i_1} = \mathsf{Enc}'(1^\lambda, pk_1, b, r_{i_1})$ and $c_{i_2} = \mathsf{Enc}'(1^\lambda, pk_2, b, r_{i_2})$
    3. invokes $\mathcal{P}(c_i, (r_{i_1}, r_{i_2}), R)$ to get p_i
  - outputs ciphertext $c = (c_1, p_1), (c_2, p_2) \dots (c_k, p_k)$
- $\mathsf{Dec}(1^\lambda, sk, c)$:
  for $i \in [k]$:
  1. invokes $\mathcal{V}(c_i, p_i, R)$ to verify if c_i is consistent
  2. if $\mathcal{V}$ accepts, computes $\hat{b_i}$ by invoking either $\mathsf{Dec}'(1^\lambda, sk_1, c_{i_1})$ or $\mathsf{Dec}'(1^\lambda, sk_2, c_{i_2})$; otherwise outputs null

4 Sketch of the security proof

We want to now show that the scheme we have described above to be secure against chosen ciphertext attacks. We want to first show that if the scheme can be broken, then either Conjecture 1 (there is a way to distinguish between $Enc(1^\lambda, pk, 0, r)$ and $Enc(1^\lambda, pk, 1, r)$) or Conjecture 2 is false ($(\mathcal{P}, \mathcal{V}, \mathcal{U})$ is not zero-knowledge).

The proof makes use of the following properties of the scheme:

- Decryption of legitimate ciphertexts requires knowledge of only one of sk_1 or sk_2.
- Given ciphertext, anyone can verify that it is a legitimate ciphertexts, e.g., a validated double encryption.

The proof attempts to conduct a CCA on the scheme to conduct a CPA on the probabilistic encryption scheme. Using an instance of the encryption scheme given in Conjecture 1 , where a public key $\langle pk_1\rangle$ which we want to distinguish between $Enc(1^\lambda, pk_1, 0, r)$ and $Enc(1^\lambda, pk_1, 1, r)$) using a regular plaintext attack, we make use of another instance of a probabilistic encryption with pk_2 and $R \in_R \mathcal{U}(|x|)$ and create a new public key $\langle pk_1, pk_2, R\rangle$. We can then simulate a CCA on this key $\langle pk_1, pk_2, R\rangle$ since we know sk_2, and then use the broken public-key $\langle pk_1, pk_2, R\rangle$ to break $\langle pk_1\rangle$

To formalize this approach, we first show in Lemma 4.2 that we can assume the message space is $\{0, 1\}$. Then, we present a procedure B that performs the simulation described above. Lemma 4.2 shows that B is well-defined using the validity and recognisability properties of the proof system (as per Theorem 3.2). Lemma 4.3 and the protocol following it demonstrate how to use B to break $\langle pk_1\rangle$ or $\langle pk_2\rangle$.

Theorem 4.1. *The scheme described in Section 3.3 is secure against CCA.*

Proof: Assuming that the scheme is breakable by machines $\mathcal{A}, \mathcal{F}$, and $\mathcal{T}$. Specifically, $\mathcal{A}$ would conduct the attack, $\mathcal{F}$ would find a pair of messages, and $\mathcal{T}$ would be able to distinguish between the pair with a certain probability. This probability ϵ turns out to be at least $\frac{1}{p(\lambda)}$ for some polynomial p for infinitely many λ's.

The proof then shows that the message space can be restricted to a single bit, and presents a standard way to do this reduction from a message space of length k (which is polynomial in λ) to a 1-bit message space.

Lemma 4.2. *If an attack on $\langle pk_1, pk_2, R\rangle$ is successful, then $\mathcal{T}$ can be used to distinguish between encryptions of 0 and 1.*

Proof: Show that if $\mathcal{F}$ finds two sequences m and m' such that the distributions of the encryptions of m and m' can be distinguished by $\mathcal{T}$, then we can use a walk on the k-cube to find a pair of neighboring vertices on the walk that differ in the jth bit.

By generating encryptions of the remaining bits in the messages m_i and m_{i+1}, we can create an encryption of either m_i or m_{i+1}, depending on whether the given bit is a $'0'$ or a $'1'$. $\mathcal{T}'s$ response is then a guess for the given bit, and the distinguishing difference is q_i. Therefore, we can assume that the message space for the challenge is {0,1}, and that $\mathcal{T}$ can distinguish with some probability $\epsilon \geq \frac{1}{p(\lambda)}$ between encryptions of 0 and 1 for some polynomial p for infinitely many λ's.

To break the scheme of Conjecture 1 with a plaintext attack, a procedure called B is presented.

A string c is invalid for R if $b_1 \neq b_2$, but $p \notin REJECT(R, c)$. Also recall that with overwhelming probability over $R \in_R \mathcal{U}$, there is no invalid word for R.

Procedure B:

Input:

- **Security Parameter:** λ
- **Encryption Keys:** pk_1, pk_2
- **Ciphertexts:** $Enc(1^\lambda, pk_1, b_1, r_1)$, $Enc(1^\lambda, pk_2, b_2, r_2)$
- **Decryption Key:** sk_1, sk_2

1. Use simulator S to generate a string R and a proof p such that $Enc(1^\lambda, pk_1, b_1, r_1)$ and $Enc(1^\lambda, pk_1, b_2, r_2)$ are consistent.
2. Pass attacker $\mathcal{A}$ the key $\langle pk_1, pk_2, R\rangle$ and simulate a CCA on it. When $\mathcal{A}$ requests for the decryption of the ciphertext, verify and then decrypt using either key sk_1 or sk_2.
3. Run $\mathcal{T}$ in attempt to decrypt $\langle Enc(1^\lambda, pk_1, b_1, r_1), Enc(1^\lambda, pk_2, b_2, r_2), p\rangle$, where $\mathcal{T}$'s guess is $t \in \{0, 1\}$

However, we also need to ensure that procedure B is well defined, such that a decryption always exists for the words that the attacker chooses, and that the outcome of B does not depend on whether sk_1 or sk_2 is known.

Lemma 4.3. *With overwhelming probability, the attacker does not ask for the decryption of an invalid word. The probability is over r_1, r_2 and the coin flips of $G, \mathcal{S}$ and $\mathcal{A}$.*

Proof: If $b_1 = b_2$, then the simulator must provide a proof for a word that belongs to the language of consistent encryptions. Theorem 3.2 reassures us that the distribution of strings R that the simulator generates is valid in this case, so the attacker $\mathcal{A}$ has only negligible probability of finding an invalid word.

But if suppose that $b_1 \neq b_2$ then $\mathcal{A}$ may have a non-negligible probability of finding an invalid word. To address this possibility, the proof refers to a claim that the following procedure M for recognising invalid words: S generated $R = R_1, R_2, ..., R_n$, half of which were generated by $\mathcal{U}$. Procedure M uses a subset J of the indices used in the proof p to check whether the corresponding R_i's were generated

by $\mathcal{U}$ or not. With overwhelming probability, all invalid words are recognized by M, and if S is given a consistent encryption, no probabilistic polynomial time machine can find a word that will be recognized as invalid by M with non-negligible probability.

Finally, the proof applies this procedure to a plaintext attack on the encryption scheme. Assuming that $\mathcal{A}$ has a non-negligible probability of finding invalid words when $b_1 = 1$ and $b_2 = 0$, the proof shows that given $Enc(1^\lambda, pk_1, b_1, r_1)$, it is possible to guess b_1 with probability better than $\frac{1}{2}$. This assumption is made without loss of generality, as one of the two inconsistent cases must have such non-negligible probability on infinitely many λ's.

1. Use G to generate pk_2 and sk_2.
2. Generate random r_2 and compute $Enc(1^\lambda, pk_2, 0, r_2)$
3. Run procedure B on input $pk_1, pk_2, Enc(1^\lambda, pk_1, b_1, r_1), Enc(1^\lambda, pk_2, 0, r_2), sk_2$; Stops if an invalid word is recognised
4. If invalid word is recognised, guess $b_1 = 1$. Else flip a coin.

Suppose if $b_1 = 1$, then there is a non-negligible probability δ that the attacker A will find an invalid word. However, with overwhelming probability, the invalid word will be recognized at step 3 of procedure B, and the probability that both events happen is at least $\frac{3}{4} * \delta$.

On the other hand, if $b_1 = 0$, then the probability of finding an invalid word or of wrongly identifying one as invalid is negligible and smaller than $\frac{\delta}{4}$. Therefore, the probability of distinguishing between encryptions of 0 and 1 is at least $\frac{\delta}{2}$ which is non-negligible.

Using Lemma 4.2, we can then define P_{00}, P_{01} and P_{11} where the output of $\mathcal{T}$ in step 3 of procedure B is $t = 1$.

The probabilities are sampled over distribution of $KeyGen(1^\lambda)$, of $Enc(1^\lambda, pk_1, b_1, \cdot)$ and $Enc(1^\lambda, pk_2, b_2, \cdot)$ as well as the coin flips of machine $\mathcal{S}, \mathcal{A}$ and $\mathcal{T}$.

Notice how these probabilities are independent of whether we are running procedure B with knowledge of sk_1 or sk_2.

Lemma 4.4. $|\mathcal{P}_{00} - \mathcal{P}_{11}| \geq \frac{\epsilon}{2}$

In order to prove this, we let $\mathcal{P}_{00}$ and $\mathcal{P}_{11}$ be the probability of output t being 1 when certifying system generates R and p. By assumption of the success of CCA, the absolute difference between $\mathcal{P}_{00}$ and $\mathcal{P}_{11}$ is greater than ϵ. If either $|\mathcal{P}_{00} - \mathcal{P'}_{00}|$ or $|\mathcal{P}_{11} - \mathcal{P'}_{11}|$ are not negligible, then we know that there exists a distinguisher D between the simulator and proof-system's output. D can then determine if this output (R, p) is generated by the simulator. This is carried out by running procedure B with R in step 1 and p in step 3, while also checking if the probability of output t being 1 is equals to $\mathcal{P}_{bb}$ in the case where (R, p) is generated by the simulator, $\mathcal{P}'_{bb}$ otherwise. Thus, we can conclude (using the assumption that the chosen-ciphertext attack is successful and the zero-knowledge property) that the absolute difference between $\mathcal{P}_{00}$ and $\mathcal{P}_{11}$ is greater than $\frac{\epsilon}{2}$.

Corollary 4.5. *either* $|\mathcal{P}_{00} - \mathcal{P}_{01} \geq \frac{\epsilon}{4}$ *or* $|\mathcal{P}_{01} - \mathcal{P}_{11} \geq \frac{\epsilon}{4}$

Assume that $|\mathcal{P}_{01} - \mathcal{P}_{11} \geq \frac{\epsilon}{4}$ for infinitely many λ's. Then consider the following procedure for a CPA on pk_1:

Procedure C:

Input:

- **Encryption Keys:** pk_1
- **Ciphertexts:** $Enc(1^\lambda, pk_1, b, r_1)$

1. Using G generate pk_2, sk_2.
2. Generate random r_2 and compute $Enc(1^\lambda, pk_2, 1, r_2)$.

3. Run procedure B on input $pk_1, pk_2, Enc(1^\lambda, pk_1, b, r_1), Enc(1^\lambda, pk_2, 1, r_2), sk_2$.

4. Output t

We know that $Pr[t = 1|b = 0] = \mathcal{P}_{01}$ and that $Pr[t = 1|b = 1] = \mathcal{P}_{11}$. From above we can also conclude that

$$Pr[t = 1|b = 0] - Pr[t = 1|b = 1] \geq \frac{\epsilon}{4}$$

giving us a distinguisher for $Enc(pk_1)$.

Specifically, if the probability difference between the output t being 1 given b is greater than a certain value (represented by $\frac{\epsilon}{4}$), then the scheme can be broken using a plaintext attack on the encryption key pk_1. This is because the attacker can distinguish between encryptions of 0 and 1, which contradicts the assumption that the scheme is secure. Therefore, pk_1 has been broken by a plaintext attack only, contradicting our assumption.

Thus this concludes the proof of the theorem, and using the results of [FLS99a] from Naor and Yung, we get the following corollary:

Corollary 4.6. *If trapdoor one-way permutations exists, then there exist CCS-PKC.*

5 Open questions

Since the same shared string R will be used for many proofs, attempt to use the Single-to-Multi-Theorem Transformations for Non-Interactive Statistical Zero-Knowledge [FR20] would have simplified somewhat the proof.

In addition, there are some following-up works that search for other assumptions than quadratic residuosity or trapdoor one-way permutation for NP language as in Conjecture 2 under which non-interactive zero-knowledge proof systems are possible such as

- Non-Interactive Zero Knowledge Proofs in the Random Oracle Model [IV19]
- Non-interactive Zero-Knowledge Proofs for NP from LWE [RSS21]
- Non-interactive Statistical Zero-Knowledge Proofs for Lattice Problems [PV08]
- Non-Interactive Zero-Knowledge Proofs for Composite Statements [AGM18]
- Multiple Non-Interactive Zero Knowledge Proofs Under General Assumptions [FLS99b]

Note that, we may still need to do some additional strengthening/conversion to ensure these non-interactive zero-knowledge proof systems satisfy necessary properties including strong soundness, validity, and recognizability as in Theorem 3.2.

References

[AGM18] Shashank Agrawal, Chaya Ganesh, and Payman Mohassel. Non-interactive zero-knowledge proofs for composite statements. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 643–673, Cham, 2018. Springer International Publishing.

[FLS99a] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29:1–28, 1999.

[FLS99b] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[FR20] Marc Fischlin and Felix Rohrbach. Single-to-multi-theorem transformations for non-interactive statistical zero-knowledge. Cryptology ePrint Archive, Paper 2020/1204, 2020. https://eprint.iacr.org/2020/1204.

[Gol93] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *J. Cryptol.*, 6(1):21–53, 1993.

[IV19] Vincenzo Iovino and Ivan Visconti. Non-interactive zero knowledge proofs in the random oracle model. Cryptology ePrint Archive, Paper 2019/952, 2019. https://eprint.iacr.org/2019/952.

[PV08] Chris Peikert and Vinod Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2008.

[RSS21] Ron Rothblum, Adam Sealfon, and Katerina Sotiraki. Toward non-interactive zero-knowledge proofs for np from lwe. *Journal of Cryptology*, 34, 01 2021.

[SMP87] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*, pages 52–72. Springer, 1987.

[Zac86] Stathis Zachos. Probabilistic quantifiers, adversaries, and complexity classes : An overview. In Alan L. Selman, editor, *Structure in Complexity Theory*, pages 383–400, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.