Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks*

Moni Naor † Moti Yung ‡

Revised July 13, 1995

Abstract

We show how to construct a public-key cryptosystem (as originally defined by Diffie and Hellman) secure against *chosen ciphertext attacks*, given a public-key cryptosystem secure against passive eavesdropping and a non-interactive zero-knowledge proof system in the shared string model. No such secure cryptosystems were known before.

Key words. cryptography, randomized algorithms

AMS subject classifications. 68M10, 68Q20, 68Q22, 68R05, 68R10

*A preliminary version of this paper appeared in the Proc. of the Twenty Second ACM Symposium of Theory of Computing.

†Incumbent of the Morris and Rose Goldman Career Development Chair, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. Work performed while at the IBM Almaden Research Center. Research supported by an Alon Fellowship and a grant from the Israel Science Foundation administered by the Israeli Academy of Sciences. E-mail: naor@wisdom.weizmann.ac.il.

‡IBM Research Division, T.J Watson Research Center, Yorktown Heights, NY 10598, USA. E-mail: moti@watson.ibm.com.

1 Introduction

Given a cryptosystem, an important (and traditional) cryptographic task is to classify the attacks which the system can withstand. The power of an attacking adversary is usually given by the information it is allowed to know about/ extract from the system. Chosen-ciphertext attack is considered to be the strongest attack on a cryptosystem. In this attack the adversary is first allowed to probe the decoding mechanism polynomially many times choosing ciphertexts of his choice getting the corresponding plaintext (messages), then without the decoding help the attacker has to break the system. (Breaking can be defined in various ways, we use the weakest existential breaking, that is getting any partial extra information about the plaintext when given the ciphertext).

A public-key cryptosystem as defined by Diffie and Hellman [8] consists of two stages. In the initiation stage, each receiver Bob publishes a public encryption key e_B (say, in some public file), while keeping secret the private decryption key d_B. In the second stage, the system is used for secure message transmissions: whoever wants to send a plaintext message M secretly to Bob, picks Bob's key e_B from the public file, encrypts the plaintext and sends him the resulting ciphertext $E_{e_B}(M)$. Only Bob can decrypt the message by applying the decryption key $D_{d_B}(E_{e_B}(M)) = M$.

Implementations of the notion were suggested by Rivest, Shamir and Adleman [32] and Merkle and Hellman [26]. The exact nature of security of these implementations was not given in a precise form, since an exact definition of security was not known at the time. Rabin [30], nevertheless, has given a scheme where an eavesdropper's ability to extract the complete message when given a ciphertext is computationally equivalent to factoring; this was the first system in which security problem was reduced to some complexity assumption (i.e., the intractability of Factoring). This presented a methodology of *reducing the security property to a well-defined complexity assumption* which, in the absence of lower bound proofs, has become the major one in cryptography.

Later, Goldwasser and Micali [19] developed the idea of *probabilistic encryption*, in which the capability to extract any partial information on the plaintext from the ciphertext, is reduced to an underlying hard problem (hard predicate). Their system which was based on the quadratic residuosity intractability assumption was the first one with this property.

The probabilistic encryption scheme of [19] and the ones that followed them were shown to withstand a *chosen-plaintext* attack, where the attacker is allowed first to use the encryption mechanism. (In a public-key cryptosystem the adversary definitely has this power.) However, it was not known whether any public key implementation is secure against chosen-ciphertext attack, where the attacker is allowed first to use the decryption mechanism as well. (A precise definition for these classical attacks is given in the next section.) Furthermore, schemes for cryptosystems which are provably secure as factoring against a chosen plaintext attack such as the Blum-Goldwasser efficient scheme for probabilistic encryption [6] are provably *insecure* against a chosen-ciphertext attack. (The same basis for its security [1] is the basis for its vulnerability to chosen ciphertext attacks.)

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. (see remark 1 in section 2.4).

We remark that chosen-ciphertext attack is also known in the folklore as the "midnight"-attack or the "lunch-break"-attack, both names describing the real-life scenario of an insider/outsider who (in the absence of others) employs the decryption mechanism, in the hope that this will help him in the cryptanalysis of future communication.

We present a method for constructing public-key cryptosystems which are secure against chosen ciphertext attacks (CCS-PKC). We use (single-theorem) non-interactive zero-knowledge proof systems for *language membership* which were introduced by Blum, Feldman and Micali [4]. In such a system for a language L the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ share a random string. $\mathcal{P}$ can prove to $\mathcal{V}$ that $x \in L$ by sending a message (which is a function of x and the shared string). Improved implementations of non interactive proof systems were done by De Santis, Micali and Persiano [7] and a better version is to be given in [5]. [4] have pointed out the potential of applying non-interactive proof systems to achieve chosen ciphertext security; this was the motivation for our investigation.

The way [4] suggested to cope with chosen ciphertext attacks was by using the notion of "non-interactive zero-knowledge proof of knowledge". However, they did not elaborate on how this notion might be implemented. Silvio Micali [27] has interpreted the claim to mean that when a user subscribes to the system he proves *interactively* possession of knowledge of a sender key, and this proof is used later on to prove non-interactively possession of knowledge of the plaintext. Thus the scheme does not fall into the category of a Diffie-Hellman PKC. To use the terminology of Micali, Rackoff and Sloan [28], this is at least a "three pass" scheme, whereas the scheme we present is "one and a half pass".

In our scheme, non-interactive zero-knowledge proofs of language membership are used in order to show the consistency of the ciphertext, but not to show that the generator of the ciphertext necessarily "knows its decryption".

The model, definitions and background are presented in section 2; the scheme and its components in section 3 and the proof of security of the scheme in section 4.

2 Background, model and definitions

2.1 Basic definitions

The notion of computationally indistinguishable distributions which originated in [19, 35] plays an important role in defining security in many cryptosystems, in particular in the context of encryptions and zero-knowledge [20, 18]. 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 2.1 *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 $x \in_R \mathcal{C}(n)$ is generated and then $\mathcal{C}$ is given the output generated by either $\mathcal{A}(x)$ or $\mathcal{B}(x)$,*

$$\begin{aligned}&|Prob[\mathcal{C}(x,y)=1|y \in_R \mathcal{A}(x)] - \\ &Prob[\mathcal{C}(x,y)=1|y \in_R \mathcal{B}(x)]| < \frac{1}{p(n)}\end{aligned}$$

for all polynomials p, for all sufficiently large n. If there exists a distinguisher $\mathcal{C}$ and an $\epsilon(n)$ such that infinitely often

$$|Prob[\mathcal{C}(x,y) = 1 | y \in_R \mathcal{A}(x)] - Prob[\mathcal{C}(x,y) = 1 | y \in_R \mathcal{B}(x)]| > \epsilon(n)$$

we say that $\mathcal{C}$ distinguishes with probability ϵ

By negligible probability we mean a probability which for any polynomial p is smaller than $\frac{1}{p(n)}$, for all but finitely many security parameters n, while an overwhelming probability is a probability which is at least $1 - \frac{1}{p(n)}$ for infinitely many n's. Thus, two ensembles of probability distribution are indistinguishable if no polynomial time machine can distinguish between them, but with negligible probability.

2.2 Public Key Cryptosystems

Definition 2.2 *A* public-key cryptosystem *consists of*

1. *A key generator G that on input n, the security parameter outputs a pair (e, d) where e is the public-key written in a public file and d is the private key.*
2. *An encryption mechanism that given a message m and the public key e produces a ciphertext c.*
3. *A decryption mechanism that on input e the public key and d the private key and a ciphertext c produces a plaintext message m.*

The components obey the following:

1. *If the decryption mechanism is given e, d, c where c was produced by the encryption mechanism on input e and m and e, d were produced by the generator, then it outputs the message m.*
2. *The generator, encryption mechanism and decryption mechanism can be operated by probabilistic machines that operate in expected polynomial in n time.*

Note that for a given key e and message m the encryption mechanism can assign one of several possibilities as a ciphertext for m.

We assume that the total number of messages sent by the cryptosystem (and their total length) is polynomial in the security parameter n.

2.3 Cryptographic attacks and security

In order to define the security level of a cryptosystem we have to specify the type of attack we are assuming (the power of the adversary) and the type of breaking which we wish to prevent (what tasks should the adversary be able to perform as the result of the attack). Given these specifications, we have to show that breaking the cryptosystem with the specified attack is as hard as performing a certain (presumed hard) computational task (such as distinguishing quadratic residues and non residues modulo a Blum integer).

The types of attacks considered in the literature are presented in an increasing order of severity:

1. ciphertext-only attack - in which the adversary sees only ciphertexts.
2. known-plaintext attack - in which the adversary knows the plaintexts (messages) and the corresponding ciphertexts transmitted.
3. chosen-plaintext (CP) attack - where the adversary gets to pick (adaptively) plaintexts of his choice and by exploiting the encryption mechanism he sees their encryption value.
4. chosen-ciphertext (CC) attack - where in addition to access to the encryption mechanism the adversary can pick (adaptively) ciphertexts of his choice and by using the decryption mechanism (as a black box) he gets the corresponding plaintexts.

In a public-key setting, a chosen plaintext attack is the weakest one which is relevant: since the encryption key is public, the adversary can definitely encrypt any plaintext of his choice.

To specify the type of breaking which we are interested in preventing, it is necessary to define the tasks that the adversary is expected be able to perform following the attack. For instance, we can consider an attack to be successful if the adversary is able to fully decrypt any ciphertext. Clearly, this requirement is too strong (that is, it does not provide enough security), since we are interested in preventing the adversary from learning partial information about the plaintext. We should note however, that even under this requirement no previous scheme was shown to be provably secure against chosen ciphertext attacks.

The notion of *semantic security*, as defined by Goldwasser and Micali, captures the requirement that it should be impossible to extract partial information on the plaintext from the ciphertext. Informally, semantic security means that any function of the plaintext that is computable with the ciphertext should be computable without it.

Though semantic security expresses the desired security properties of cryptosystems a notion that is easier to prove is *indistinguishability of encryptions* (a.k.a. polynomial security). These two notions were shown to be equivalent by Goldwasser and Micali [19] (indistinguishability $\Rightarrow$ semantic), Micali, Rackoff and Sloan [28] (semantic $\Rightarrow$ indistinguishability) and Goldreich [14] (the uniform case).

Under the notion of indistinguishability of encryptions, the cryptosystem is considered to have been broken if the adversary can find two messages m_0 and m_1 in the message space such that it can distinguish between encryptions of m_0 and m_1. This requirement implies that in the public-key setting the encryption must be probabilistic, i.e. $E(m_1)$ and $E(m_2)$ are probability distributions that are indistinguishable to those who do not know the decryption key. From now on we identify the notion of security with that of indistinguishability of encryptions.

We now, following [13], define precisely what we mean by a chosen ciphertext attack: an attack consists of three probabilistic polynomial time machines $\mathcal{A}$, $\mathcal{F}$, $\mathcal{T}$. Each machine corresponds to a different stage of the attack.

A *(participating) CC-attacker* is a probabilistic machine $\mathcal{A}$ whose input is a security parameter n (in unary), and a public key e drawn from I_n. $\mathcal{A}$ is allowed to query the decryption machine (which knows the key d corresponding to e) with any ciphertext polynomially many times. Each time it gets as a result a plaintext corresponding to the input ciphertext. $\mathcal{A}$'s computation is allowed to be a function of the input, the queries and answers and of its random bits. $\mathcal{A}$ produces some output, which w.l.o.g. can be the entire history of its computation and its internal state (which are polynomial in n).

The second machine $\mathcal{F}$, the *message-finder*, has as input the public key, the security parameter n and an auxiliary input (which may contain the entire history of $\mathcal{A}$'s computation plus its inner state). $\mathcal{F}$ produces as output two messages m_0, m_1.

The third machine T, the *message-distinguisher*, gets as its input the security parameter, the public key, the computation history of $\mathcal{F}$ which includes the pair of messages m_0, m_1, the inner state of $\mathcal{F}$ and a challenge which is an encryption of one of the two messages m_0, m_1. It outputs one bit which is its guess as to which of the two messages was encrypted.

A chosen plaintext attack is defined similarly, except that the participating stage $\mathcal{A}$ does not have access to the decryption mechanism, but has access to the encrypting mechanism. (This is trivially achieved in a public-key cryptosystem).

Definition 2.3 *Consider the following experiment which we call an XX-attack. Run the algorithm G on input 1^n to produce a pair of keys (e, d) (d is kept secret at the decoding machine). Then run a participating XX-attacker, with input e to produce a history h. Next, give the public key e and h as input to $\mathcal{F}$ in order to obtain a pair of messages $m_0, m_1 \in M_n$ and the attack history h; choose one of these at random, m_b say, and give (m_b, e) to E to produce an encryption α; finally, give (e, m_0, m_1, h, α) as input to T to obtain the bit b'. The experiment is a success if $b = b'$; otherwise it is a failure.*

XX can stand for either CC or CP resulting in a CC-attack or CP-attack respectively.

Definition 2.4 *We say that PKC (E, D) is* secure against XX-attack *if for any polynomial p, for any XX-attacker $\mathcal{A}$, for any XX-message-finder $\mathcal{F}$, for any message-distinguisher $\mathcal{T}$, for all sufficiently large n:*

$$|Prob[\text{success}] - Prob[\text{failure}]| < \frac{1}{p(n)}.$$

We have defined the attack in terms of Turing machines. Alternatively, we could have chosen a non-uniform model such as circuits. The results of the paper remain the same in this case, i.e. a non-uniform assumption yields a non-uniform scheme.

2.4 Background Remarks

Before we continue we close the section by some background and essential clarifying remarks on related work.

Remark 1: Since CCS-PKC were not known, a totally different mode of operation was used to achieve CC secure message transmission: interactive protocols (telephone conversations), rather than PKC (mail sending). In interactive protocols the parties exchange messages, and indeed the solutions suggested by Goldwasser, Micali and Tong [22], Yung [36] and Galil, Haber and Yung [13] were all inherently interactive. (This, of course, did not solve the open problem regarding PKC). The first two were given without the exact notion of security and used exchange of new cryptographic keys via interaction. The third one uses "interactive proof-systems of knowledge" as was formalized by Feige, Fiat and Shamir and

Tompa and Woll [11, 34]. The sender proves that she knows the ciphertext and thus the CC-attack is reduced to chosen-plaintext one. As mentioned above, Micali [27] has clarified that the claims about chosen ciphertext secure cryptosystems made in section 5 of [4] refer to a system with initial interaction as well.

Remark 2: More is known about security of signatures than CCS-PKC: Goldwasser, Micali and Rivest [21] have defined a hierarchy of attacks similar to the one given here. They showed how to implement a signature scheme secure against the strongest attack on signature schemes (adaptive chosen message attack) with the weakest notion of breaking (existentially forgeable) under the assumption that factoring is hard (or that claw-free trapdoors exist). Later, Bellare and Micali [3] showed how to base signatures on any trapdoor permutation. Naor and Yung [29] then showed how to construct a trapdoorless signature, basing it on what they called universal one-way hash functions which they implemented using any 1-1 one-way function. Recently, Rompel [33] has shown how to construct universal one-way hash functions from any one-way function.

Remark 3: Bellare and Goldwasser [2] have shown how to get signatures from non interactive zero-knowledge proof systems combined with the pseudo-random functions of Goldreich, Goldwasser and Micali [15]. Their scheme required a stronger definition of non interactive zero knowledge proof system that is not known to be equivalent to the one used here. Nevertheless it can be implemented under QRA.

Remark 4: We use a cryptosystem design principle which can be called *secret hiding*: design the system such that it has two "secrets". In order to operate it, only one of the secrets needs to be known. However, to an outsider it should be indistinguishable which of the secrets is known. This principle was introduced by Feige and Shamir [10].

3 The scheme

In our scheme the public key of a user consists of three parts: two keys e_1, e_2 of an encryption function E and a random string R generated by some distribution. To send a message in this scheme, it should be encrypted according to each encryption key and a proof should be appended that the same message was encrypted in both keys. The proof would be in a non-interactive zero-knowledge proof system with a shared random string R. When decrypting a message, it is first verified in the non-interactive proof system that the two encryptions are consistent, and only then is the message decrypted.

In section 3.1 we define the basic tools needed for the scheme: a probabilistic encryption scheme secure against chosen plaintext attacks and (single theorem) non-interactive zero-knowledge proof systems in the shared string model. The non-interactive proof systems as defined by Blum, Feldman and Micali [4] are not strong enough for our purposes. Our scheme actually requires it to satisfy additional properties. In section 3.2 we show how to transform any non-interactive zero-knowledge proof system so as to satisfy these additional requirements. Finally in section 3.3 we present the scheme itself.

3.1 Basic tools

Probabilistic encryption

For our scheme we require the following assumption

Assumption 1 *There is a triple* (E, D, G) *with the following properties*

- G*, the* key-generator*, is a probabilistic machine that, given the security parameter* n*, halts in expected time polynomial in* n *and outputs a pair of strings* (e, d)*. (*e *a* public key *and* d *a* secret key*).*
- E*, the encryption function, gets 3 inputs:* e *which is the first output of* G*,* $b \in \{0,1\}$ *and* r *which is a random string of length* $p(n)$ *where* p *is a polynomial.* $E_e(b,r)$ *is computable in polynomial time.*
- D*, the decryption function, gets two inputs:* c *which is a ciphertext and a private key* d *which was produced by* G*.* $D_d(c)$ *is computable in expected polynomial time*
- *if* G *outputs* (e, d)*, then*

$$\forall b \in \{0,1\},\ \forall r \in \{0,1\}^{p(n)} \quad D_d(E_e(b,r)) = b$$

- *Indistinguishability - for all polynomial time machines* M

$$\begin{aligned}&|Prob[M(e, E_e(0,r)) = 1] - \\ &\quad Prob[M(e, E_e(1,r)) = 1]| < \frac{1}{poly(n)}\end{aligned}$$

  where the probability is taken over the coin flips of G, E *and* M*.*

For implementations of probabilistic encryption see [19, 1, 6, 35, 28]. From the hard core predicate results of Yao [35], Levin [25] and Goldreich and Levin [17] it follows that if there are public-key cryptosystems which are secure in any reasonable sense (i.e. if the plaintext is chosen at random, then it is hard to completely retrieve it given only its ciphertext), then schemes with the properties of assumption 1 exist. In particular, if trapdoor permutations exists, then such schemes exist.

Non-interactive zero-knowledge proof systems

We define non-interactive proofs as in [4]. A (single theorem) non-interactive proof system for a language L allows one party $\mathcal{P}$ to prove membership in L to another party $\mathcal{V}$ for any $x \in L$. $\mathcal{P}$ and $\mathcal{V}$ initially share a string R of length polynomial in the security parameter n. To prove membership of a string x in $L_n = L \cap \{0,1\}^n$, $\mathcal{P}$ 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 $\mathcal{U}(n)$ that can be generated by a probabilistic polynomial time machine. (In all the examples we know of it is the uniform distribution, though it is required for our scheme.)

Let L be in NP. For any $x \in L$ there is a set of witnesses for its membership. Let $WL(x) = \{z |\ z\ is\ a\ witness\ for\ x\}$. For the proof system to be of any use, $\mathcal{P}$ should be able to operate in polynomial time if it given a witness $z \in WL(x)$. In general z is not available to $\mathcal{V}$.

Let $\mathcal{P}(x, z, R)$ be the distribution of the proofs that P generates on input x, witness z and shared string R. Suppose that $\mathcal{P}$ sends $\mathcal{V}$ a proof p when the shared random string is R. Then the pair (R, p) is called the conversation. Any $x \in L$ and $z \in WL(x)$ induces a probability distribution $\mathcal{CONV}(x,z)$ of conversations (R,p) where $R \in \mathcal{U}$ is a shared string and $p \in \mathcal{P}(x,z,R)$ is a proof.

For the system to be zero-knowledge, it should have a simulator $\mathcal{S}$. $\mathcal{S}$ on input x generates a conversation (R,p). Let $\mathcal{S}(x)$ be the distribution of the conversation that S generates on input x, and let $\mathcal{S}_R(x)$ be the distribution of the R part of the conversation.

Let $ACCEPT(R,x) = \{p | \mathcal{V} \text{ accepts on input } R, x, p\}$,
and let $REJECT(R,x) = \{p | \mathcal{V} \text{ rejects on input } R, x, p\}$.

The following is the definition of non-interactive proof systems of [4] which is modified to incorporate the tractability of $\mathcal{P}$. The uniformity conditions of the system are adopted from Goldreich [14].

Definition 3.1 *A triple $(\mathcal{P},\mathcal{V},\mathcal{U})$ where $\mathcal{P}$ is a probabilistic machine, $\mathcal{V}$ is a polynomial time machine and $\mathcal{U}$ is a polynomial time sampleable probability distribution is a* non-interactive zero-knowledge proof system *for the language $L \in NP$ if:*

1. Completeness: *(if $x \in L$ then $\mathcal{P}$ generates a proof that $\mathcal{V}$ accepts). For all $x \in L_n$, for all $z \in WL(x)$, with overwhelming probability for $R \in_R \mathcal{U}(n)$ and $p \in_R \mathcal{P}(x,z,R)$, p is in $ACCEPT(R,x)$. The probability is over the choices of the shared string R and the inner coin flips of $\mathcal{P}$.*
2. Soundness: *(if $y \notin L$ then no prover can generate a proof that $\mathcal{V}$ accepts). For all $y \notin L_n$ with overwhelming probability over $R \in_R \mathcal{U}(n)$ for all $p \in \{0,1\}^*$, p is in $REJECT(R,y)$. The probability is over the choices of the shared string R.*
3. Zero-knowledge: *(There is a probabilistic polynomial time machine $\mathcal{S}$ which is a simulator for the system). For all probabilistic polynomial time machines $\mathcal{C}$, if $\mathcal{C}$ generates $x \in L$ and $z \in WL(x)$ then,*

$$\begin{aligned} &|Prob[\mathcal{C}(w) = 1 | w \in_R \mathcal{S}(x)] - \\ &Prob[\mathcal{C}(w) = 1 | w \in_R \mathcal{CONV}(x,z)]| < \frac{1}{p(n)} \end{aligned}$$

for all polynomial p and sufficiently large n.

Assumption 2 *For all $L \in NP$ there is a non-interactive proof system as defined above.*

Currently this assumption is known to be true assuming the intractability of quadratic residuosity [4, 7] or given any trapdoor one-way permutation [12]. Recently, Kilian and Petrank [23, 24] found more efficient implementations of such schemes. Their scheme is for the circuit satisfiability problem. The length of a proof (and the size of the shared random string) of a satisfiable circuit of size L, assuming trapdoor permutation on k bits, is $O(Lk^2)$. Remark: Note that though the same R will be used for many proofs we do not require the multi-theorem version of non-interactive zero-knowledge proof systems (which we have not defined). Using the Bellare and Goldwasser [2] definition of multi-theorem zero-knowledge would have simplified somewhat the proof. However, we do not know if the two definitions are equivalent in the sense that the existence of the single theorem type implies the existence of the Bellare-Goldwasser type.

3.2 Strengthening non-interactive proof systems

For the non-interactive proof system to be useful for our purposes it must satisfy more requirements. We will show how to transform any non-interactive zero-knowledge proof system into one that satisfies those requirements. In the transformations applied to the proof system we utilize the fact that we can run several proof systems simultaneously, while maintaining the zero-knowledge property. This *parallel composition* is not known to be true in interactive proof systems (see Goldreich and Krawczyk [16] for evidence against it).

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. Let $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ be constructed as follows: $\mathcal{U}$ generates a string $R = R_1, R_2 \ldots, R_n$ such that $R_i \in_R \mathcal{U}_i(n)$ for $1 \leq i \leq n$. On input x and $z \in WL(x)$ and $R = (R_1, R_2, \ldots, R_n)$ $\mathcal{P}$ generates $p = (p_1, p_2, \ldots, p_n)$ where $p_i \in_R \mathcal{P}_i(x, z, R_i)$ for $1 \leq i \leq n$. Verifier $\mathcal{V}$, on input $p = (p_1, p_2 \ldots p_n)$, x , $R = (R_1, R_2, \ldots, R_n)$, runs $\mathcal{V}_i$ on x, R_i, p_i for all $1 \leq i \leq n$. $\mathcal{V}$ 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: It is not hard to see that completeness and soundness are preserved. Suppose that there is a distinguisher $\mathcal{C}$ for for the system. I.e. $\mathcal{C}$ can generate an $x \in L$ and $z \in WL(x)$ and can distinguish with probability ϵ between

$$(\mathcal{CONV}_1(x, z), \mathcal{CONV}_2(x, z), \ldots, \mathcal{CONV}_n(x, z))$$

and

$$(\mathcal{S}_1(x), \mathcal{S}_2(x), \ldots \mathcal{S}_n(x))$$

From this it follows that there is $1 \leq i \leq n$ such that $\mathcal{C}$ can distinguish with probability at least $\frac{\epsilon}{n}$

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

and

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

By definition of non-interactive proof system $\mathcal{CONV}_j$ and $\mathcal{S}_j$ are polynomial time sampleable. Thus, $\mathcal{C}$ can distinguish between $\mathcal{S}_i$ and $\mathcal{CONV}_i$ with probability $\frac{\epsilon}{n}$. $\Box$.

Strong soundness

The first requirement is that the soundness condition will hold even if x is chosen after R is known. (This is required since R is given as part of the public-key, and x, which should be a consistent double encryption, is selected by the sender afterwards.) Strong soundness requirement is: with overwhelming probability over $R \in_R \mathcal{U}(n)$, $\forall y \notin L_n$, $\forall p \in \{0,1\}^*$, $p \in REJECT(R, y)$. In order to satisfy this requirement, a quantifier swapping technique of Zachos [37] can be used: For words of length n, R would actually be a sequence of $2n$ strings $R_1, R_2, \ldots R_{2n}$, where each $R_i \in_R \mathcal{U}(n)$. 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, run $\mathcal{V}$ on each proof p_i. Accept if for all $1 \le i \le 2n$ $\mathcal{V}$ accepted. By Lemma 3.1 the resulting scheme is still a non-interactive zero-knowledge proof system.

As for strong soundness, for any $y \notin L_n$ and any $1 \le i \le 2n$ the probability that there exists a proof p_i such that $p_i \notin REJECT(R_i, y)$ is small. Thus the probability that there are proofs $p_i \notin REJECT(R_i, y)$ for all $1 \le i \le 2n$ is smaller than 2^{-2n}. Thus, for $R = R_1, R_2 \ldots R_{2n}$ where $R_i \in_R \mathcal{U}(n)$ the probability that there exists a $y \notin L$ of length n such that for all $1 \le i \le 2n$ there is a $p_i \notin REJECT(R_i, y)$ is smaller than 2^{-n}.

We can therefore assume that the soundness condition holds even if x is chosen after $R = R_1, R_2 \ldots R_{2n}$ is known.

Valid distributions

A harder requirement to satisfy is what we call *validity.* By the strong soundness we know 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 $\mathcal{U}$. Nevertheless, we would like it to be impossible to find a $y \notin L$ and p such that $p \notin REJECT(R, y)$. One cannot argue that if false statements are proved using the output of the simulator, then we have a distinguisher between $Q_R(x)$ and $\mathcal{U}(|x|)$ (and thus a distinguisher between $\mathcal{S}(x)$ and $\mathcal{CONV}(x, z)$), since we might not recognize that a false statement is being proved.

Definition 3.2 *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 $p \in ACCEPT(y, R)$.*

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

What we would like to achieve is that if $x \in L$ then it should be hard 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 3.3 *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 $\{\mathcal{Q}_R(x) | x \in L\}$ 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. The distribution $\mathcal{U}$ is valid.

We construct a system $(\mathcal{P}', \mathcal{V}', \mathcal{U}')$ such that the distribution $\mathcal{S}'_R$ is valid. In the new system, for security parameter n, R is a sequence $R_1, R_2, \ldots R_n$. (Each $R_i \in_R \mathcal{U}(|x|)$ is, as above, in itself a sequence, assuring strong soundness.) To prove that $x \in L$, $\mathcal{P}'$ picks a random subset $J \subset \{1 \ldots n\}$ of size $\frac{n}{2}$. For each $i \in U$ 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 $1 \le j \le \frac{n}{2}$ $\mathcal{V}$ accepted; $ACCEPT'(R, x)$ is defined accordingly.

The simulator $\mathcal{S}'$ for the new proof system works by letting $\mathcal{S}$, the simulator for the old system, run $\frac{n}{2}$ times and in addition generating $\frac{n}{2}$ strings by distribution $\mathcal{U}(n)$. 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. By an argument similar to Lemma 3.1 the resulting system is still a non-interactive zero-knowledge proof system for L.

We will show that the distribution $\mathcal{S}'_R(x)$ obeys the requirements specified above.

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: Suppose 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. Clearly, $q_0 = \frac{1}{\binom{n}{n/2}}$ and by assumption $q_{n/2} > \delta$. Thus, there is some $1 \leq i < \frac{n}{2}$ such that $q_{i+1} - q_i > \frac{\delta}{n}$.

Given a string R for which it should be decided whether it is the output of $\mathcal{S}_R$ or of $\mathcal{U}$ do the following: add it 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. $\Box$

Claim 3.2 *The distribution S'_R is valid*

Proof: Suppose that 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). Thus $\mathcal{M}$ can be used to separate the outputs of S from the random ones and by the previous claim to distinguish between $\mathcal{S}(x)$ and $\mathcal{CONV}(x, z)$. $\Box$.

The following claim implies 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 $\mathcal{S}$. Thus, the first property is satisfied. By the previous claim ($\mathcal{S}'_R$ is valid) it is clear that if $x \in L$, with overwhelming probability $\mathcal{C}$ will not output a word that is recognized as invalid by this method. □

Call the properties defined in the claim above *recognizability.*

Thus we have:

Theorem 3.2 *Any non-interactive zero-knowledge proof system for a language L can be converted to one with the following properties.*

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

We call a proof system with those properties a *certifying system.*

3.3 The scheme

Definition 3.4 *Let (E, D, G) be as in assumption 1. For public keys e_1, e_2 a* consistent double encryption *is $w = E_{e_1}(b, r_1), E_{e_2}(b, r_2)$ for some $b \in \{0,1\}$, $r_1, r_2 \in \{0,1\}^{p(n)}$.*

The language of consistent double encryptions

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

is in NP. For a given word $w = E_{e_1}(b, r_1), E_{e_2}(b, r_2)$ the pair $r_1,\ r_2$ is a witness for its membership in L. By assumption 2 and Theorem 1 there is a certifying system $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ for L.

We are now ready to present the scheme

Key generation:
Run $G(n)$ twice to generate (e_1, d_1) and (e_2, d_2). Generate $R \in_R \mathcal{U}(n)$.
Public key is $\langle e_1, e_2, R\rangle$. Private key is $\langle d_1, d_2\rangle$.

Encryption:
To encrypt a message $m = b_1, b_2, \ldots b_k$, with public key $\langle e_1, e_2, R\rangle$:
for each $1 \leq i \leq k$

1. generate $r_{i_1}, r_{i_2} \in_R \{0,1\}^{p(n)}$
2. compute $c_i = E_{e_1}(b, r_{i_1}), E_{e_2}(b, r_{i_2})$
3. run $\mathcal{P}$ on c_i with witness (r_{i_1}, r_{i_2}) and string R to get p_i.

The encrypted message is $(c_1, p_1), (c_2, p_2) \ldots (c_k, p_k)$.

Decryption:
To decrypt a ciphertext $(c_1, p_1), (c_2, p_2) \ldots (c_k, p_k)$:
for all $1 \leq i \leq k$

1. verify that c_i is consistent by running the verifier $\mathcal{V}$ on c_i, p_i, R.
2. If $\mathcal{V}$ accepts, then retrieve b_i by computing either $D_{d_1}(E_{e_1}(b_i, r_{i_1}))$ or $D_{d_2}(E_{e_2}(b_i, r_{i_2}))$. Otherwise the output is null.

From this description it is clear that the generator and the encryption and decryption mechanism can be operated in polynomial time. Also if the decryption mechanism is given a legitimate ciphertext and the right key it produces the message encrypted.

4 Proof of security

We now show that the scheme described in the previous chapter is secure against chosen ciphertext attacks. We will show that if it can be broken, then either assumption 1 is false, specifically there is a way to distinguish between $E_e(0, r)$ and $E_e(1, r)$, or assumption 2 is false, namely the system $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ is not zero-knowledge.

The proof utilizes the following properties of the scheme

- Decryption (of legitimate ciphertexts) requires knowledge of only one of d_1 or d_2.
- Given a ciphertext, any user can verify that it is a legitimate ciphertext, i.e., a validated double encryption.

The idea of the proof is to use the chosen ciphertext attack on the scheme to conduct a chosen plaintext attack on the probabilistic encryption scheme. Given an instance of the encryption scheme from assumption 1, i.e. a key $\langle e_1 \rangle$ for which we wish to distinguish between $E_{e_1}(0, r)$ and $E_{e_1}(1, r)$, using a plaintext attack, we generate another instance of the probabilistic encryption e_2 and $R \in_R \mathcal{U}(|x|)$ and create a public key (in our scheme) $\langle e_1, e_2, R \rangle$. We now simulate a chosen ciphertext attack on $\langle e_1, e_2, R \rangle$. We can simulate such an attack since we know d_2 and from the properties listed above, this suffices. It is then shown how to use the broken public-key $\langle e_1, e_2, R \rangle$ in order to break $\langle e_1 \rangle$.

In order to formalize this intuition we first prove in Lemma 4.2 that we can assume that our message space is $\{0, 1\}$. Then we present procedure B that performs the simulation described above. Lemma 4.2 shows that B is well defined using the validity and recognizability of the proof system (implied by Theorem 1). Lemma 4.3 and the protocol following it show how to use B in order to break $\langle e_1 \rangle$ or $\langle e_2 \rangle$.

Theorem 4.1 *The scheme described in section 3.3 is secure against CC-attack.*

Proof: If the scheme can be broken, then there are machines $\mathcal{A}$, $\mathcal{F}$ and $\mathcal{T}$ such that $\mathcal{A}$ conducts the participating chosen ciphertext attack, $\mathcal{F}$ finds a pair of messages and $\mathcal{T}$ can distinguish between the pair with probability ϵ where ϵ is at least $\frac{1}{p(n)}$ for some polynomial p for infinitely many n's.

We first show that we can restrict ourselves to the single bit message version. We present a standard reduction for any such n from a fixed $k = poly(n)$ bit long message space to a 1-bit long messages.

Lemma 4.2 *If an attack on $\langle e_1, e_2, R\rangle$ is successful, then $\mathcal{T}$ can be used to distinguish between encryptions of 0 and 1.*

Proof: For $\langle e_1, e_2, R\rangle$ to have been broken means that $\mathcal{F}$ has found two sequences $m = b_1, b_2, \ldots b_k$ and $m' = b'_1, b'_2, \ldots b'_k$ such that the distributions of the encryptions of m and m' can be distinguished by the machine $\mathcal{T}$ with some non-negligible probability q (i.e. q is the difference in the probability that the distinguisher $\mathcal{T}$ outputs a '1' when given an encryption of m and m'. Consider a walk on the k-cube that connects vertex m and m', $m = m_1, m_2, \ldots m_\ell = m'$, $(l \leq k)$. For each neighboring pair of vertices on the walk, m_i and m_{i+1}, let q_i be the difference of the probabilities that the distinguisher outputs '1' given an encryption of m_i and m_{i+1}. $q = \sum_{i=1}^{\ell} q_i$. This implies $q \leq \sum_{i=1}^{\ell} |q_i|$. Hence there must be some i such that $|q_i| \geq \frac{q}{\ell}$. Say that m_i and m_{i+1} differ in the jth bit.

To distinguish between encryptions of '0' and '1' the following can be done: Given an encryption of a bit, generate encryptions of the rest of the matching bits in the messages m_i and m_{i+1}; compose them to create an encryption of word which is either m_i or m_{i+1}, depending if the given bit is a '0' or a '1'. Now the distinguisher's response is a guess for the given bit. The distinguishing difference is q_i. $\Box$

Therefore w.l.o.g., the message space for the challenge is $\{0,1\}$ and we assume that a successful CC-attack is possible, thus $\mathcal{T}$ can distinguish with some probability $\epsilon \geq \frac{1}{p(n)}$ between encryptions of 0 and 1 for some polynomial p for infinitely many n's. From now on we assume that n is in this infinite set.

Recall that a string $c = E_{e_1}(b_1, r_1), E_{e_2}(b_2, r_2), p$ is an *invalid word* for R if $b_1 \neq b_2$ and yet $p \notin REJECT(R, c)$. Recall that with overwhelming probability over $R \in_R \mathcal{U}$ there is no invalid word for R.

We now present procedure B, a *gedanken* (thought) experiment which is the keystone in breaking the scheme of assumption 1 $\langle E\rangle$ with a plaintext attack. Assume that e_1 and e_2 were generated by G and r_1 and r_2 are random.

Procedure B:
Input: security parameter n, encryption keys e_1 , e_2, ciphertexts $E_{e_1}(b_1, r_1)$, $E_{e_2}(b_2, r_2)$ and either decryption key d_1 or d_2.

1. Using the simulator S generate a string R and a proof p that $E_{e_1}(b_1, r_1)$ and $E_{e_2}(b_2, r_2)$ are consistent.

2. Give the attacker $\mathcal{A}$ the key $\langle e_1, e_2, R\rangle$ and simulate a CC-attack on it. When $\mathcal{A}$ asks for the decryption of a ciphertext, verify its consistency and then decrypt using whichever key is known, d_1 or d_2.

3. Run $\mathcal{T}$ in attempt to decrypt $\langle E_{e_1}(b_1, r_1),\ E_{e_2}(b_2, r_2),\ p\rangle$. Say that $\mathcal{T}$'s guess is $t \in \{0,1\}$.

Before we proceed, we must verify that procedure B is well defined, i.e., that a decryption always exists for the words the attacker chooses (we haven't defined the decryption of an invalid word), and that the outcome of B does not depend on whether d_1 or d_2 are 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 has to provide a proof for a word which is in the language of consistent encryptions. We are assured by Theorem 1 that the distribution of strings R that the simulator generates is valid in this case and hence the attacker $\mathcal{A}$ has only negligible probability of finding an invalid word.

Suppose that if $b_1 \neq b_2$ then $\mathcal{A}$ has a non-negligible probability of finding an invalid word. Recall from Claim 3.3 the following procedure M for recognizing invalid words: $\mathcal{S}$ generated $R = R_1, R_2, \ldots R_n$, half of which were generated by $\mathcal{U}$. Given a proof p, check whether the subset $J \subset \{1, .., n\}$ of indices used in the proof p includes exactly the R_i's that were not generated by $\mathcal{U}$. The recognizability property is that with overwhelming probability (over the choices of R) all invalid words are recognized by M and that, if $\mathcal{S}$ is given a consistent encryption then no probabilistic polynomial time machine can find with non-negligible probability a word that will be recognized as invalid by M.

We will use this procedure for a plaintext attack on E. Assume that when $b_1 = 1$ and $b_2 = 0$ the attacker $\mathcal{A}$ has a non-negligible probability δ of finding invalid words; (this is w.l.o.g., since under the assumption one of the two inconsistent cases must have such non-negligible probability on infinitely many n's). given $E_{e_1}(b_1, r_1)$ we want to guess b_1 with probability better than $\frac{1}{2}$.

1. Using G generate e_2 and d_2.
2. Generate random r_2 and compute $E_{e_2}(0, r_2)$.
3. Run B on input $e_1, e_2,\ E_{e_1}(b_1, r_1), E_{e_2}(0, r_2),\ d_2$;
   Stop if an invalid word is recognized (by the method described above).
4. If an invalid word was recognized, guess that $b_1 = 1$. Otherwise flip a coin.

If $b_1 = 1$ there is non-negligible probability δ that an invalid word is found by $\mathcal{A}$. With overwhelming probability the invalid word is recognized at step 3, and the probability that both events happen is at least $\frac{3}{4} \cdot \delta$. If $b_1 = 0$, then the probability of finding an invalid word or of erroneously identifying one as such is negligible and smaller than $\frac{\delta}{4}$. Thus the probability of distinguishing between encryptions of '0' and '1' is at least $\frac{\delta}{2}$ which is non-negligible. $\square$

The significance of Lemma 4.2 is that we can now define P_{00}, P_{01} and P_{11} as the probabilities that the output of $\mathcal{T}$ in step 3 of B is $t = 1$, given that $b_1 = b_2 = 0$, $0 = b_1 \neq b_2 = 1$, and $b_1 = b_2 = 1$ respectively. The probabilities are taken over the distributions of G, of $\langle E_{e_1}(b_1, \cdot\rangle$ and $\langle E_{e_2}(b_2, \cdot\rangle$ and the coin flips of the machines $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{T}$. It is easy to see that these probabilities are independent of whether we are running B knowing d_1 or d_2 (up to a negligible factor).

Lemma 4.4 $|P_{00} - P_{11}| > \frac{\epsilon}{2}$

Proof: Let P'_{00} and P'_{11} be the probability that the output t is 1 in case R used in step 1 and p used in step 3 are generated by the certifying system and not by the simulator, i.e.

$R \in_R \mathcal{U}(n)$ and $p \in_R \mathcal{P}((E_{e_1}(b,r_1), E_{e_2}(b,r_2)), (r_1,r_2), R)$. By assumption of the success of the CC- attack $|P'_{00} - P'_{11}| > \epsilon$. If $|P_{00} - P'_{00}|$ or $|P_{11} - P'_{11}|$ are not negligible, then we have a distinguisher between the simulator and the proof-system's output: given

$$(E_{e_1}(b,r_1), E_{e_2}(b,r_2)),\ (e_1,e_2),\ (d_1,d_2),\ R,\ p$$

for which we wish do check if (R,p) are the output of $\mathcal{S}$ or not, we run B with R in step 1 and p in step 3. The probability that $t = 1$ is P_{bb} in case $(R,p) \in_R \mathcal{S}$ and P'_{bb} otherwise.

Thus (by the assumption that CC- attack is successful and the zero-knowledge property) we can conclude that $|P_{00} - P_{11}| > \frac{\epsilon}{2}$. □

Corollary 4.1 *either* $|P_{00} - P_{01}| > \frac{\epsilon}{4}$ *or* $|P_{01} - P_{11}| > \frac{\epsilon}{4}$.

Assume w.l.o.g. that $|P_{01} - P_{11}| > \frac{\epsilon}{4}$ for infinitely many n's (which is the set of n's under attack). Consider the following procedure for a chosen plaintext attack on $\langle e_1 \rangle$:
Input $\langle e_1 \rangle$, $E_{e_1}(b,r_1)$

1. Using G generate e_2, d_2.
2. generate random r_2 and compute $E_{e_2}(1,r_2)$.
3. run procedure B on input $e_1, e_2,\ E_{e_1}(b,r_1), E_{e_2}(1,r_2),\ d_2$.
4. output t, ($\mathcal{T}$'s guess from step 3 of B).

We know that $Pr[t = 1|b = 0] = P_{01}$ and that $Pr[t = 1|b = 1] = P_{11}$. From the above we can conclude that

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

and thus we have a distinguisher for E_{e_1}. In other words $\langle e_1 \rangle$ has been broken by a plaintext attack only, contradicting our assumption. This concludes the proof of the theorem □

Using the results of [12, 5] we have:

Corollary 4.5 *If trapdoor one-way permutations exists, then there exist CCS-PKC.*

5 Conclusions and extensions

We have shown how to construct a public key system secure against chosen ciphertext attacks from probabilistic encryption schemes and non-interactive zero-knowledge proof systems. This motivates searching for more general assumptions (than quadratic residuosity) under which non-interactive zero-knowledge proof system are possible. In particular, our work suggests looking for non-interactive proofs of consistency that are not necessarily good for all languages in NP.

Recently, Dolev, Dwork and Naor have introduced a new primitive called non-malleable bit commitment. In the public-key setting, this primitive yields a scheme where an eavesdropper cannot create a ciphertext whose corresponding plaintext is a function of the messages that were sent in the system (except for exact copying). This in conjunction with our CCS-PKC implies a cryptosystem that is secure against an even stronger attack than CC, suggested by Rackoff and Simon [31]: the attacker gets to use the decryption mechanism even after seeing the ciphertext he is interested in cracking. The only restriction is that he won't use the decryption mechanism on exactly the same ciphertext he wants to crack.

Acknowledgments

We thank Manuel Blum, Uri Feige, Oded Goldreich, Stuart Haber, Russell Impagliazzo, Charlie Rackoff and Adi Shamir for valuable discussions. We thank Oded Goldreich for many remarks leading to improvements of this paper.

References

[1] W. Alexi, B. Chor, O. Goldreich and C. Schnorr, *RSA/Rabin Bits are* $1/2 + 1/poly$ *Secure*, Siam Journal on Computing, 17(2) (1988), pp.194-209.

[2] M. Bellare and S. Goldwasser, *New Paradigms for Digital Signatures and Message Authentication based on Non-interactive Zero-knowledge Proofs*, Crypto 89.

[3] M. Bellare and S. Micali, *How to Sign Given Any Trapdoor Function*, Proc. 20th Annual Symposium on the Theory of Computing, Chicago, 1988, pp.32-42.

[4] Blum M., P. Feldman and S. Micali, *Non-Interactive Zero-Knowledge Proof Systems*, Proc. 20th Annual Symposium on the Theory of Computing, Chicago, 1988, pp 103-112.

[5] M. Blum, A. De Santis, S. Micali and , G. Persiano, *Non-Interactive Zero-Knowledge*, Manuscript.

[6] M. Blum and S. Goldwasser, *An Efficient Probabilistic Public-key Encryption that Hides All Partial Information*, Proc. of Crypto 84, pp. 289-299.

[7] A. De Santis, S. Micali and G. Persiano, *Non-Interactive Zero-Knowledge Proof Systems* Proc. of Crypto 87.

[8] W. Diffie and M. Hellman, *New Directions in Cryptography* , IEEE Trans. on Information Theory 22(6), 1976, pp. 644-654.

[9] D. Dolve, C. Dwork and M. Naor, *Non-malleable cryptography,*

[10] U. Feige and A. Shamir, *Witness Hiding and Witness Indistinguishability*, STOC 1990.

[11] U. Feige, A. Fiat and A. Shamir, *Zero Knowledge Proofs of Identity*, J. of Cryptology 1 (2), pp 77-94.

[12] U. Feige, D. Lapidot and A. Shamir, *Multiple Non-Interactive Zero-Knowledge Proofs Based on a Single Random String*, Proceedings of 31st Symposium on Foundations of Computer Science, 1990, pp. 308-317.

[13] Z. Galil, S. Haber and M. Yung, *Interactive Public-key Cryptosystems*, Submitted to J. of Cryptology, (preliminary version in Crypto 85).

[14] O. Goldreich, *A Uniform Complexity Encryption and Zero-knowledge*, Technion CS-TR 570, June 1989.

[15] O. Goldreich S. Goldwasser and S. Micali, *How to Construct Random Functions* , J. of the ACM 33 (1986), pp. 792-807.

[16] O. Goldreich and H. Krawczyk, *On the Composition of Zero-knowledge Proof Systems*, Technion CS-TR 568, June 1989. (To appear ICALP 90).

[17] O. Goldreich and L. Levin, *A Hard Predicate for All One-way Functions* , Proc. 21st Annual Symposium on the Theory of Computing, Seattle, 1989, pp. 25-32.

[18] S. Goldreich, S. Micali and A. Wigderson, *Proofs that Yield Nothing But their Validity, and a Methodology of Cryptographic Protocol Design*, Proceedings of the 27th Symposium on the Foundation of Computer Science, 1986, pp. 174-187.

[19] S. Goldwasser and S. Micali, *Probabilistic Encryption*, J. Com. Sys. Sci. 28 (1984), pp 270-299.

[20] S. Goldwasser, S. Micali and C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, Siam J. on Computing, 18(1) (1989), pp 186-208.

[21] S. Goldwasser, S. Micali and R. Rivest, *A Secure Digital Signature Scheme* , Siam Journal on Computing, Vol. 17, 2 (1988), pp. 281-308.

[22] S. Goldwasser, S. Micali and P. Tong, *Why and How to Establish a Private Code on a Public Network*, Proceedings of the 23rd Symposium on the Foundation of Computer Science, 1982, pp. 134-144.

[23] J. Kilian, *On the complexity of bounded-interaction and non-interactive zero-knowledge proofs*, Proceedings of the 35th IEEE Symposium on the Foundation of Computer Science, 1994.

[24] J. Kilian and E. Petrank, *An efficient non-interactive zero-knowledge proof system for NP with general assumptions*, Electronic Colloquium on Computational Complex ity TR95-038. Available: http://www.eccc.uni-trier.de/eccc/info/ECCC.

[25] L. Levin, *One-way Functions and Pseudo-random Generators*, Combinatorica 7 (1987), pp. 357-363.

[26] R. Merkle and M. Hellman, *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Trans. on Information Theory, vol. IT-24, 5 (1978), pp.525-530.

[27] S. Micali, Personal Communication, February 1990.

[28] S. Micali and C. Rackoff and R. Sloan, *Notions of Security of Public-Key Cryptosystems*, SIAM J. on Computing 17(2) 1988, pp. 412-426.

[29] M. Naor and M. Yung, *Universal One-way Hash Functions and their Cryptographic Applications*, Proc. 21st Annual Symposium on the Theory of Computing, Seattle, 1989, pp. 33-43.

[30] M. O. Rabin, *Digital Signatures and Public Key Functions as Intractable as Factoring*, Technical Memo TM-212, Lab. for Computer Science, MIT, 1979.

[31] C. Rackoff and D. Simon, Manuscript, 1990.

[32] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signature and Public Key Cryptosystems*, Comm. of ACM, 21 (1978), pp 120-126.

[33] J. Rompel, *One-way Function are Necessary and Sufficient for Signatures*, STOC 1990.

[34] M. Tompa and H. Woll, *Random Self Reducibility and Zero-knowledge Interactive Proofs of Knowledge*, Proceedings of the 28th Symposium on the Foundation of Computer Science, 1987.

[35] A. C. Yao, *Theory and Applications of Trapdoor functions*, Proceedings of the 23th Symposium on the Foundation of Computer Science, 1982, pp. 80-91.

[36] M. Yung, *Cryptoprotocols: Subscription To a Public Key*, Proceedings of Crypto 84, Springer-Verlag, 1985, pp 439-453.

[37] S. Zachos, *Probabilistic Quantifiers, Adversaries and Complexity Classes: an overview*, Proceeding of Structure in Complexity, Springer Verlag, 1986, pp. 383-400.