NON-INTERACTIVE ZERO-KNOWLEDGE PROOF SYSTEMS.

Alfredo De Santis †
Silvio Micali *
Giuseppe Persiano †

Abstract.

The intriguing notion of a Zero-Knowledge Proof System has been introduced by Goldwasser, Micali and Rackoff [GMR] and its wide applicability has been demonstrated by Goldreich, Micali and Wigderson [GMW1]-[GMW2].

Based on complexity theoretic assumptions, Zero-Knowledge Proof Systems exist, provided that

(i) The prover and the verifier are allowed to talk back and forth.

(ii) The verifier is allowed to flip coins whose result the prover cannot see.

Blum, Feldman and Micali [BFM] have recently shown that, based on specific complexity theoretic assumption (the computational difficulty of distinguishing products of two primes from those product of three primes), both the requirements (i) and (ii) above are not necessary to the existence of Zero-Knowledge Proof Systems. Instead of (i), it is enough for the prover only to talk and for the verifier only to listen. Instead of (ii), it is enough that both the prover and verifier share a randomly selected string.

We strengthen their result by showing that Non-Interactive Zero-Knowledge Proof Systems exist based on the weaker and well-known assumption that quadratic residuosity is hard.

† Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84100 Salerno, Italy.

* MIT, Laboratory for Computer Science, Cambridge, Mass. 02139. Supported by NSF grant DCR-8413577.

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 52-72, 1988.

1. Introduction.

In many scenarios, like cryptographic ones, "knowledge" is the most valuable resource and thus one may not want to give away more knowledge than "absolutely needed".

This has been formalized by Goldwasser, Micali and Rackoff [GMR] who introduced the somewhat paradoxical notion of a *Zero-Knowledge Proof System (ZKPS)*. Since then other definitions of Zero-Knowledge has been given by [GHY] and [BC].

This is a special way of proving theorems. It allows A, who holds the proof of a given theorem T, to convince a poly-bounded B that T is true without revealing any additional information. In other words, in a ZKPS A allows B to verify that a theorem T is true; however, he does not allow B, after verifying the proof, to compute more than he could have computed after being only told (without any proof) that T was indeed true.

Under the assumption that one-way functions exist, Goldreich, Micali and Wigderson [GMW1] show that membership in any language in NP can be proved in Zero-Knowledge. ZKPS is indeed a powerful notion and have had a strong impact in the field of cryptographic protocols; see [FFS], [FMRW] and most notably, the recent completeness theorem for protocols with honest majority of [GMW2].

1.1 The communication model needed for Zero-Knowledge.

ZKPS's have been defined and have been shown to exist for a particular communication model between "prover" and "verifier": the interactive Turing machine model of [GMR]. The salient features of this model of communication are the following two:

(i) *Interaction.* The prover and the verifier are allowed to talk back and forth.
(ii) *Privacy.* The verifier is allowed to flip coins whose result the prover cannot see.

Though interaction and privacy of computation are requirements that can be met, in practice they may not be readily available. For example this is the case if A will be away for 10 years and the mail is the only way to communicate with B. During this trip, each time A will succeed in proving a new theorem, he may wish to prove it in Zero-Knowledge to B by simply sending B a letter to which B need not to reply. Thus

Is A's wish possible?

This question has been addressed by Blum, Micali and Feldman [BMF]. Rephrasing it in a more theoretical way:

What are the essential communication features
that make Zero-Knowledge proofs possible?

Assuming that quadratic residuosity is hard, an assumption widely used in Cryptography [GM], we show that ZKPS's exist provided that

the prover and the verifier share
(i.e. have as input a common)
random enough string.

The communication model consisting of sharing a random string was proposed in [BFM]. They showed that in this model ZKPS's exist if it is difficult to distinguish numbers product of two primes from those product of three primes. Our result improves on theirs as the Quadratic Residuosity Assumption is weaker than the assumption used in [BFM]. In fact, if it is easy to distinguish a quadratic residue from a quadratic non residue then it is also easy to distinguish a number product of two primes from one product of three primes, while the converse is not known to be true. Our result is also an improvement in that our algorithm is simpler.

Notice that sharing a random string is a weaker communication model than the one proposed by [GMR]. In fact, in the [GMR] model, prover and verifier may agree on a random enough string by using a coin-flipping protocol [B]. Actually, the new communication model, in a sense explained in section 3.1, is the *minimal* one supporting ZKPS's.

The new model dispenses with the need of privacy of coin tosses as the verifier does not need to use more randomness than the one contained in the common string. Moreover, it dispenses with the need of interaction as well. Once the prover and the verifier share a random string σ, to prove that $x \in L \in NP$, it is enough for the prover to compute a proof (string), $Proof_{x,\sigma}$, in a special way on input x and σ, and send it to the verifier. The verifier, computing on input σ, x and $Proof_{x,\sigma}$ will correctly check that $x \in L$ while receiving zero additional knowledge.

Let us rephrase this in terms of our previous example, where the prover A was leaving for a long trip. Here, if A and B, before A leaves for his long trip, have witnessed some random events, (e.g. a lottery, the sun stains, ...) or have consulted the RAND tables, then A can prove a theorem in Zero-Knowledge just by writing a letter to which B need not to reply.

Let us now proceed more formally.

2. Preliminaries.

2.1 Notations and conventions.

Let us quickly recall the standard notation of [GoMiRi].

We emphasize the number of inputs received by an algorithm as follows. If algorithm A receives only one input we write "$A(\cdot)$", if it receives two inputs we write "$A(\cdot,\cdot)$" and so on.

If $A(\cdot)$ is a probabilistic algorithm, then for any input x, the notation $A(x)$ refers to the probability space that assigns to the string σ the probability that A, on input x, outputs σ. If S is a probability space, then $Pr_S(e)$ denotes the probability that S associates with the element e.

If $f(\cdot)$ and $g(\cdot,\ldots,\cdot)$ are probabilistic algorithms then $f(g(\cdot,\ldots,\cdot))$ is the probabilistic algorithm obtained by composing f and g (i.e. running f on g's output). For any inputs $x, y, \ldots$ the associated probability space is denoted by $f(g(x,y,\ldots))$.

If S is any probability space, then $x \leftarrow S$ denotes the algorithm which assigns to x an element randomly selected according to S. If F is a finite set, then the notation $x \leftarrow F$ denotes the algorithm which assigns to x an element selected according to the probability space whose sample space is F and uniform probability distribution on the sample points.

The notation $Pr\big(x \leftarrow S; y \leftarrow T; \ldots : p(x,y,..)\big)$ denotes the probability that the predicate $p(x,y,\ldots)$ will be true after the ordered execution of the algorithms $x \leftarrow S$, $y \leftarrow T, \ldots$

The notation $\{x \leftarrow S; y \leftarrow T; \ldots : (x,y,\ldots)\}$ denotes the probability space over $\{(x,y,\ldots)\}$ generated by the ordered execution of the algorithms $x \leftarrow S$, $y \leftarrow T, \ldots$.

Let us recall the basic definitions of [GMR]. We address the reader to the original paper for motivation, interpretation and justification of these definitions.

Let $U = \{U(x)\}$ be a family of random variables taking values in $\{0,1\}^*$, with the parameter x ranging in $\{0,1\}^*$. $U = \{U(x)\}$ is called poly-bounded family of random variables, if, for some constant $e \in \mathcal{N}$, all random variables $U(x) \in U$ assign positive probability only to strings whose length is exactly $|x|^e$.

Let $C = \{C_x\}$ be a poly-size family of Boolean circuits, that is, for some constants $c, d > 0$, all C_x have one Boolean output and at most $|x|^c$ gates and $|x|^d$ inputs. In the

following, when we say that a random string, chosen according to $U(x)$, where $\{U(x)\}$ is a poly-bounded family of random variables, is given as input to C_x, we assume that the length of the strings that are assigned positive probability by $U(x)$ equals the number of boolean inputs of C_x.

Definition (Indistinguishability). *Let $L \subset \{0,1\}^*$ be a language. Two poly-bounded families of random variables $U = \{U(x)\}$ and $V = \{V(x)\}$ are indistinguishable on L if for all poly-size families of circuits $C = \{C_x\}$,*

$$\left| Pr\big(a \leftarrow U(x) : C_x(a) = 1\big) - Pr\big(a \leftarrow V(x) : C_x(a) = 1\big) \right| < |x|^{-O(1)}$$

with $x \in L$.

Definition (Approximability). *Let $L \subset \{0,1\}^*$ be a language. A family of random variables $U = \{U(x)\}$ is approximable on L if there exists a Probabilistic Turing Machine M, running in expected polynomial time, such that the families $\{U(x)\}$ and $\{M(x)\}$ are indistinguishable on L.*

2.2 Number theory.

Let $\mathcal{N}$ denote the natural numbers, $x \in \mathcal{N}$, $Z_x^* = \{y \mid 1 \leq y < x, \ \ gcd(x,y) = 1\}$ and $Z_x^{+1} = \{y \in Z_x^* | (y \mid x) = +1\}$, where $(y \mid x)$ is the Jacobi symbol. We say that $y \in Z_x^*$ is a quadratic residue modulo x iff there is $w \in Z_x^*$ such that $w^2 \equiv y \bmod x$. If this is not the case we call w a quadratic non residue modulo x.

Define the quadratic residuosity predicate to be

$$\mathcal{Q}_x(y) = \begin{cases} 0, & \text{if } y \text{ is a quadratic residue modulo } x; \\ 1, & \text{otherwise;} \end{cases}$$

and the languages QR and QNR as

$$QR = \{(y,x) | \mathcal{Q}_x(y) = 0\}$$

$$QNR = \{(y,x) | y \in Z_x^{+1} \text{ and } \mathcal{Q}_x(y) = 1\}.$$

Fact 1: For each $y_1, y_2 \in Z_x^{+1}$ one has

$$\mathcal{Q}_x(y_1 y_2) = \mathcal{Q}_x(y_1) \oplus \mathcal{Q}_x(y_2).$$

Let $Z_s(n)$ denote the set of n-bit integers product of $s \geq 1$ distinct primes. In the following we will use the:

Quadratic Residuosity Assumption (QRA). *For each poly-size family of circuits* $\{C_n \mid n \in \mathcal{N}\}$,

$$Pr\big(x \leftarrow Z_2(n); y \leftarrow Z_x^{+1}: C_n(x,y) = \mathcal{Q}_x(y)\big) < 1/2 + 1/n^{-O(1)}.$$

The QRA is widely used in Cryptography, see for example [GM].

The current fastest algorithm to compute $\mathcal{Q}_x(y)$ is to first factor x and then compute $\mathcal{Q}_x(y)$, while it is well known that, given the factorization of x, $\mathcal{Q}_x(y)$ can be computed in $O(|x|^3)$ steps. Therefore it is usual to choose x product of two large primes of the same length since these integers constitute the hardest input for a factoring algorithm.

3. The main results.

To prove the existence of Non-Interactive Zero-Knowledge Proof Systems for all NP languages, it is enough to prove it for the NP-complete language $3SAT$ [GJ]. For $k > 0$, we define the language $3SAT_k = \{x \in 3SAT \mid |x| \leq k\}$.

We present our result first in a weaker form, for simplicity sake.

3.1 A first solution.

Here we show that if a n^3-bit long string is randomly selected and given to both parties, then the prover can show that any *single* $x \in 3SAT_n$ is indeed satisfiable. Notice that this is not as general as what we promised in the introduction. Only in section 3.2 we will show that, for each polynomial $Q(\cdot)$, using the *same* randomly chosen n^3-bit long σ, any $Q(n)$ formulae $x_1, \ldots, x_{Q(n)} \in 3SAT_n$ can be shown to be satisfiable in Zero-Knowledge.

In the following we formally define the Single-Theorem Non-Interactive ZKPS.

Definition. *A Single-Theorem Non-Interactive ZKPS is a pair (A,B) where A is a Probabilistic Turing Machine and* $B(\cdot,\cdot,\cdot)$ *is a deterministic algorithm running in time polynomial in the length of its first input, such that:*

1) **Completeness.** *(The probability of succeeding in proving a true theorem is overwhelming.)*

$\exists c > 0$ *such that* $\forall x \in 3SAT_n$

$$Pr(\sigma \leftarrow \{0,1\}^{n^c}; y \leftarrow A(\sigma,x): B(x,y,\sigma) = 1) > 1 - n^{-O(1)}.$$

2) **Soundness.** *(The probability of succeeding in proving a false theorem is negligible.)* $\exists c > 0$ *such that* $\forall x \notin 3SAT_n$ *and for each Probabilistic Turing Machine* A'

$$Pr(\sigma \leftarrow \{0,1\}^{n^c}; y \leftarrow A'(\sigma, x): B(x, y, \sigma) = 1) < n^{-O(1)}.$$

3) **Zero-Knowledge.** *(The proof gives no information but the validity of the theorem.)* $\exists c > 0$ *such that the family of random variables* $V = \{V(x)\}$ *where*

$$V(x) = \{\sigma \leftarrow \{0,1\}^{|x|^c}; y \leftarrow A(\sigma, x): (\sigma, y)\}$$

is approximable over $3SAT$.

Comment. As promised in the introduction we give evidence that our model is the minimal supporting Zero-Knowledge.

The simplest communication model is certainly NP. In it the verifier is deterministic and thus needs not to worry about the privacy of his (non existent) coin tosses. Moreover the interaction is absolutely elementary: on input $x \in L \in NP$, the prover has to talk only once and the verifier has only to listen. What the prover says is w_x, a witness that $x \in L$. Unfortunately, such simplicity does not support Zero-Knowledge. It is not hard to prove that if a language L possesses an NP Proof System that is Zero-Knowledge, then $L \in P$.

Consider a probabilistic version of NP in which the verifier checks that w_x is a proper witness in probabilistic poly-time. Then it is easy to prove that if L possessed such a proof system that is Zero-Knowledge, L would belong to BPP.

So if the prover and the verifier are completely independent, Zero-Knowledge proofs are not possible. In order to have Zero-Knowledge proofs, the prover and the verifier must have something random in common and the simplest way to do this is sharing a random string.

In the following we exhibit a Single-Theorem Non-Interactive ZKPS. We first present informally our protocol and then we prove the following

Theorem 1. *Under the QRA, there exists a Single-Theorem Non-Interactive ZKPS (A,B).*

An informal view of our protocol.

Let $\sigma = \sigma_0 \dots \sigma_{n^3+7n^2-1}$ be a random string shared by A and B.

Let $C = \{c_1, c_2, \dots, c_m\}$, $C \in 3SAT_n$, a collection of clauses on the set $U = \{u_1, u_2, \dots, u_k\}$ of boolean variables. A wants to prove to B that there is a truth assignment that satisfies all the clauses in C, without yielding any additional information. Let $t: U \to \{T, F\}$ be a truth assignment satisfying C.

A's proof consists of two parts: first he exhibits a pair (x, y) such that $x \in Z_2(n)$ and $(y, x) \in QNR$, then he proves $C \in 3SAT$.

A chooses a random $x \in Z_2(n)$, and divides the first $7n^2$ bits of σ in $7n$ n-bit strings, thus obtaining $7n$ integers. A discards those integers not in Z_x^{+1} and partitions the remaining integers in two equivalence classes (one formed of residues and one formed of non residues). To show that two integers, a and b, are in the same class, that is $\mathcal{Q}_x(a) = \mathcal{Q}_x(b)$, A exhibits a random square root modulo x of ab.

A then, chooses two random elements, y_0, y_1, one from each class, if any, and computes $y = y_0 y_1 \bmod x$. Note that $\mathcal{Q}_x(y_0) \neq \mathcal{Q}_x(y_1)$ and so, by Fact 1, y is a non quadratic residue modulo x.

B is now convinced that $x \in Z_2(n)$ and $(y, x) \in QNR$.

A associates an element $w_j \in Z_x^{+1}$ to each literal u_j in such a way that w_j is a non quadratic residue modulo x iff u_j is true under t. To make this association, for each variable u_j, A chooses a random $r_j \in Z_x^*$. Then if u_j is true under t, A associates $r_j^2 \bmod x$ to the literal u_j and $yr_j^2 \bmod x$ to $\overline{u}_j$. If u_j is false under t, A associates $yr_j^2 \bmod x$ to the literal u_j and $y^2 r_j^2 \bmod x$ to $\overline{u}_j$. At this point B can check that, for each variable, the value associated to u_j is a quadratic residue iff the value associated to $\overline{u}_j$ is a non quadratic residue and so he is sure that exactly one of the two literals u_j and $\overline{u}_j$ is true under t.

In this way a triple of values is naturally associated to each clause and A has to prove that each triple is not formed by 3 quadratic residues, that is each clause is satisfied by t.

A divides the last n^3 bits of σ in n sets formed of n strings of n-bit, thus obtaining n sets of n integers. Each set is associated to a clause.

For each clause, A discards from the associated set the integers not in Z_x^{+1} and groups the remaining integers in triple. Since each element of a triple can be a quadratic residue or not, the triples obtained can be partitioned in 8 equivalence classes, that must be roughly

of the same size since $x \in Z_2(n)$.

A computes the partition and shows that it is correct in the following way. To show that the tern (a,b,c) is in the same class as (d,e,f), A exhibits random square roots modulo x of ad, be, cf. B can check the correctness of the step, but does not know which kind of triples are in any equivalence class. A can show the class consisting of the terns formed by three quadratic residues by simply disclosing their square roots modulo x.

Finally, let (h,l,m) be the tern associated to the clause c_i. A shows that (h,l,m) belongs to one of the remaining classes, by disclosing three square roots as done above. B can easily check the correctness of the last step. Moreover B knows nothing but that the triple (h,l,m) is not formed by three quadratic residues.

Proof (of theorem 1).

We start by formally describing the protocol (A,B).

A's protocol.

When we say "A writes τ", we mean that A appends τ followed by a special symbol, such as #, to the string $Proof$ that will be sent to B.

Let $\sigma = \sigma_0 \ldots \sigma_{n^3+7n^2-1}$.

Let $C = \{c_1, c_2, \ldots, c_m\}$, $C \in 3SAT_n$, a collection of clauses on the set $U = \{u_1, u_2, \ldots, u_k\}$ of boolean variables. A wants to prove B that there is a truth assignment that satisfies all the clauses in C, without yielding any additional information. Let $t: U \to \{T,F\}$ be a truth assignment satisfying C.

1) A sets $Proof$ = empty string and writes C.

2) A chooses a random $x \in Z_2(n)$ and writes it.
   A sets E_0, E_1= empty set.

3) For $i = 1, \ldots, 7n$. Let s_i be the integer whose binary representation is $\sigma_{(i-1)n} \cdots \sigma_{in-1}$. If $s_i \notin Z_x^{+1}$ then A discards s_i. If in E_j, $j \in \{0,1\}$, there is an element s such that $(ss_i, x) \in QR$ then A writes (j, s, γ), where γ is a random square root modulo x of ss_i, and puts s_i in E_j. Otherwise A puts s_i in one of the empty sets, E_j, and writes $(j, 0, s_i)$.

4) If E_0 or E_1 (or both) is empty then A halts.

Otherwise, A randomly chooses $y_0 \in E_0$ and $y_1 \in E_1$, computes $y = y_0 y_1 \bmod x$ and writes (y, y_0, y_1).

5) A picks at random $r_j \in Z_x^*$, $j = 1, \ldots, k$.
If $t(u_j) = F$ A sets $w_j = r_j^2 \bmod x$, otherwise A sets $w_j = y r_j^2 \bmod x$. A writes $(w_1, w_2, \ldots, w_k)$.

6) For each clause c_i, $i = 1, \ldots m$, A performs steps 6.1-6.6.

6.1) Let the clause c_i consist of literals $z_{i_1}, z_{i_2}, z_{i_3}$. If the literal z_{i_j}, $j = 1,2,3$, is the variable u_l then A sets $\eta_{i_j} = w_l$. If the literal z_{i_j} is the complement of the variable u_l then A sets $\eta_{i_j} = y w_l \bmod x$.

6.2) A sets ST = empty stack and repeats n times step 6.3.

6.3) For $j = 0, \ldots, n-1$. Let z be the element whose binary representation is $\sigma_{7n^2+(i-1)n^2+(j-1)n} \cdots \sigma_{7n^2+(i-1)n^2+jn-1}$.
If $z \notin Z_x^{+1}$ then A discards z, otherwise A puts z in the stack ST.

6.4) A sets E_s = empty set, $s = 1, \ldots, 7$.
A repeats step 6.5 until the number of elements in the stack ST is less than 3.

6.5) A picks three elements $\alpha_1, \alpha_2, \alpha_3$ from the stack ST. If $\alpha_1, \alpha_2, \alpha_3$ are quadratic residue modulo x, then A sets γ_j = a random square root of α_j modulo x, $j = 1,2,3$, and writes $(0, \alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2, \gamma_3)$. Otherwise, if there is a triple $(\beta_1, \beta_2, \beta_3)$ in the set E_s, $1 \le s \le 7$, such that $\alpha_1\beta_1$, $\alpha_2\beta_2$, $\alpha_3\beta_3$ are quadratic residue modulo x then A puts $(\alpha_1, \alpha_2, \alpha_3)$ in the set E_s, sets γ_j = a random square root of $\alpha_j\beta_j$ modulo x, $j = 1,2,3$, and writes $(s, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3)$; otherwise A puts $(\alpha_1, \alpha_2, \alpha_3)$ in the empty set E_s, $1 \le s \le 7$, with smallest index (i.e. $1 \le j < s$ implies E_j not empty) and writes $(s, 0, 0, 0, 0, 0, 0)$.

6.6) If there is a triple $(\beta_1, \beta_2, \beta_3)$ in the set E_s, such that $\eta_{i_1}\beta_1, \eta_{i_2}\beta_2, \eta_{i_3}\beta_3$ are quadratic residues modulo x, A sets γ_j = a random square root of $\eta_{i_j}\beta_j$ modulo x, $j = 1,2,3$, and writes $(s, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3)$.

B's protocol.

1) B reads the collection $C = \{c_1, \ldots, c_m\}$ of clauses and sets E_0, E_1= empty set.

2) For $i = 1, \ldots, 7n$, let s_i the integer whose binary representation is $\sigma_{in-n} \ldots \sigma_{in-1}$. If $s_i \notin Z_x^{+1}$ then B discards s_i. Otherwise B reads the triple (d_1, d_2, d_3). If $d_2 \neq 0$ then

B checks that $s_i d_2 \equiv d_3^2 \bmod x$, that $d_2 \in E_{d_1}$ and puts s_i in E_{d_1}. If $d_2 = 0$ then B checks that E_{d_1} is empty and puts s_i in E_{d_1}.

3) B checks that E_0 and E_1 are not empty. B reads (y, y_0, y_1) and checks that $y = y_0 y_1 \bmod x$ and that $y_0 \in E_0$ and $y_1 \in E_1$.

4) B reads $(w_1, w_2, \ldots, w_k)$. B checks that $w_j \in Z_x^{+1}$, $j = 1, 2, \ldots, k$.

5) For each clause c_i, $i = 1, \ldots, m$, B performs steps 5.1-5.5.

5.1) B sets ST = empty stack and repeats n times step 5.2.

5.2) For $j = 0, \ldots, n-1$. Let z be the element whose binary representation is $\sigma_{7n^2+(i-1)n^2+(j-1)n} \cdots \sigma_{7n^2+(i-1)n^2+jn-1}$.
If $x \notin Z_x^{+1}$ then B discards z, otherwise B puts z in the stack ST.

5.3) B sets E_s = empty set, $s = 1, \ldots, 7$. B repeats step 5.4 until the number of elements in the stack ST is less than 3.

5.4) B picks three elements $\alpha_1, \alpha_2, \alpha_3$ from the stack ST. B reads the 7-tuple $(k_0, \ldots, k_6)$ from the letter. If $k_0 = 0$ then B checks that $\alpha_j = k_{j+3}^2 \bmod x$, $j = 1, 2, 3$. Otherwise if $k_0 = s$, $1 \leq s \leq 7$, and E_s is not empty B checks that $(k_1, k_2, k_3) \in E_s$ and that $k_j \alpha_j \equiv k_{j+3}^2 \bmod x$, $j = 1, 2, 3$, and then puts $(\alpha_1, \alpha_2, \alpha_3)$ in E_s. Otherwise B checks that $k_0 = s$, $1 \leq s \leq 7$, E_s is empty, and that E_j is not empty for $j = 1, \ldots, s-1$ and then puts $(\alpha_1, \alpha_2, \alpha_3)$ in E_s.

5.5) If each E_s, $s = 1, \ldots, 7$, is not empty then B reads the 7-tuple $(k_0, \ldots, k_6)$ from the letter. Let the clause c_i consists of literals $z_{i_1}, z_{i_2}, z_{i_3}$. If the literal z_{i_j}, $j = 1, 2, 3$, is the variable u_l then B sets $\eta_{i_j} = w_l$. If the literal z_{i_j}, $j = 1, 2, 3$, is the complement of the variable u_l then B sets $\eta_{i_j} = y w_l \bmod x$. B checks that $k_0 = s$, $1 \leq s \leq 7$, $(k_1, k_2, k_3) \in E_s$ and that $k_j \eta_{i_j} \equiv k_{j+3}^2 \bmod x$, $j = 1, 2, 3$.

If all the checks are successful B stops and accepts, otherwise B rejects.

In the following we prove Theorem 1 by showing that the above protocol is a Single-Theorem Non-Interactive ZKPS.

(A,B) satisfies the Completeness requirement.

Say that B has received a proof that the collection of clause $C = \{c_1 \ldots c_m\}$ over the set $U = \{u_1 \ldots u_k\}$ of boolean variables is satisfiable.

Assume that C is indeed satisfiable, let t be a truth assignment that satisfies C and suppose that A and B follow the specification of the protocol.

Let p_x, $x \in \mathcal{N}$, be the probability that a random y, $\mid y \mid \leq \mid x \mid$, is in Z_x^*, i.e. $p_x = Pr\big(y \leftarrow \{z | 0 \leq z < 2^{|x|}\} : y \in Z_x^*\big)$. Notice that $p_x > 2/5$, for all $x \in Z_2(n)$ and sufficiently large n.

A halts at step 4 only when among the $7n$ integers obtained from σ those in Z_x^{+1} are all quadratic residues or all non quadratic residues modulo x. Denote with Q_x the probability of this event. One has that Q_x satisfies

$$Q_x \leq 2\Big(1 - \frac{p_x}{4}\Big)^{7n}, \tag{1}$$

and so is negligible.

Consider the relation $\stackrel{R}{=}$ on the set $Z_x^{+1} \times Z_x^{+1} \times Z_x^{+1}$ defined as:

$$(\alpha_1, \alpha_2, \alpha_3) \stackrel{R}{=} (\beta_1, \beta_2, \beta_3) \quad \text{iff} \quad (\alpha_i \beta_i, x) \in QR,\ i = 1, 2, 3.$$

It can be easily seen that $\stackrel{R}{=}$ is an equivalence relation and that the quotient set $(Z_x^{+1} \times Z_x^{+1} \times Z_x^{+1})/\stackrel{R}{=}$ consists of 8 equivalence classes, $EQ_0, \ldots, EQ_7$. Let EQ_0 be the equivalence class formed of triple constituted of 3 quadratic residues.

If A follows the protocol properly then all the non empty sets E_s, $1 \leq s \leq 7$, are subsets of different equivalence classes. Furthermore all the triples that A does not put in any set E_s, $1 \leq s \leq 7$, at step 6.5 (that is the triples formed by 3 quadratic residues) are all in EQ_0. Therefore if A follows the protocol then he can always perform step 6.5 and all the checks made by B at step 1-5.5 are successful.

Consider now the clause c_i and let $\eta_{i_1}, \eta_{i_2}, \eta_{i_3}$ be the values computed at step 6.1 by A. Two cases are possible :

i) The sets E_s, $s = 1, \ldots, 7$ are non empty. Thus there is a set, E_r, such that each $(\beta_1, \beta_2, \beta_3) \in E_r$ is equivalent to $(\eta_{i_1}, \eta_{i_2}, \eta_{i_3})$. A can prove such an equivalence by simply showing $(\beta_1, \beta_2, \beta_3)$ and exhibiting random square roots modulo x of $\eta_{i_1}\beta_1, \eta_{i_2}\beta_2, \eta_{i_3}\beta_3$.

ii) Some of the E_s, $1 \leq s \leq 7$, are empty and $(\eta_{i_1}, \eta_{i_2}, \eta_{i_3})$ is not equivalent to any triple in the non empty sets.

From the above discussion we conclude that if A follows the protocol and doesn't halt at step 4, B always accepts. Therefore, the Completeness requirement of Single-Theorem Non-Interactive ZKPS's is met.

(A,B) satisfies the Soundness requirement.

B accepts a non satisfiable collection of clauses C either if

i) A cheated him during the proof that $x \in Z_2(n)$ and $(y,x) \in QNR$.

ii) $x \in Z_2(n)$ and $(y,x) \in QNR$ but A cheated him in proving that each clause is satisfied.

We now show that the probabilities of i) and ii) are negligible.

Suppose $x \in Z_s(n)$, $s > 2$. Consider the equivalence relation $\stackrel{\circ}{=}$ defined over Z_x^{+1} as

$$y_1 \stackrel{\circ}{=} y_0 \iff (y_1 y_2, x) \in QR.$$

Z_x^{+1} is partitioned by the relation $\stackrel{\circ}{=}$ in 2^{s-1} equivalence classes. The only case in which B accepts x is when among the $7n$ integers obtained from σ, those with Jacobi symbol $+1$ belong to at most two equivalence classes. Let P_s denote this probability. From

$$P_s \leq \binom{2^{s-1}}{2}\left(1 - \frac{2^{s-1} - 2^{s-2} - 1}{2^{s-1}} p_x\right)^{7n} + \binom{2^{s-1}}{1}\left(1 - \frac{2^s - 2^{s-1} - 1}{2^s} p_x\right)^{7n}$$

and $s < n$ we obtain that P_s is negligible.

The only case in which A can make B accept $(y,x) \notin QNR$ is when among the $7n$ integers obtained from σ those in Z_x^{+1} are all quadratic residues or all non quadratic residues modulo x. Note that when A chooses x, he already knows σ and so he could choose x such that this event occurs. So we have to show that the probability that, given $7n$ randomly chosen n-bit integers, $z_1, \ldots, z_{7n}$, there exists a $x \in Z_2(n)$ such that those integers $z_i \in Z_x^{+1}$ are all quadratic residues or all non quadratic residues modulo x is negligible. The probability, $Q(n)$, of this event is, from (1),

$$Q(n) \leq \sum_{x \in Z_2(n)} Q_x \leq 2^{n+1}\left(\frac{9}{10}\right)^{7n}.$$

Thus the probability of i) is negligible.

Now suppose that $x \in Z_2(n)$ and $(y,x) \in QNR$.

The only case in which ii) occurs is when there is an equivalence class, EQ_s, $1 \leq s \leq 7$, such that no triple of it is in the stack ST during the proof that a clause c_i is satisfied. Indeed if there is a triple for each EQ_s, $1 \leq s \leq 7$, then each E_s is not empty and if the clause c_i is not satisfied by the truth assignment t, A at step 6.6 cannot prove $(\eta_{i_1}, \eta_{i_2}, \eta_{i_3})$ to be equivalent to any triple in any of the sets E_s, $s = 1, ..., 7$.

On the other hand, if there is a EQ_s such that no triple of it is in the stack ST then A could split the set of triples formed of 3 quadratic residues in two parts. Then he could reveal that one of these two parts is formed of 3 quadratic residue by sending $(0, \ldots)$ along with the 3 square roots modulo x and finally could put the remaining triples in the empty set E_s. In this way A could show the equivalence of $(\eta_{i_1}, \eta_{i_2}, \eta_{i_3})$ with an element in such a set.

From the above discussion, we can say that the probability that B accepts a non satisfiable C, in the case ii), is no greater than the probability that during one of the n iterations of steps 6.1-6.5 at least one of the equivalence classes EQ_s, $s = 1, \ldots, 7$, has no triple in the stack ST. Let R denote this event. Then

$$\begin{aligned} Pr(R) &= \sum_{j=0}^{\lfloor n/3 \rfloor} Pr(R \mid \text{ there are } j \text{ triples in ST }) Pr(\text{there are } j \text{ triples in ST }) \\ &= \sum_{j=0}^{\lfloor n/3 \rfloor} \left(\frac{7}{8}\right)^j Pr(\text{ there are } j \text{ triples in ST }). \end{aligned}$$

The probability that there are l elements in the stack is $\binom{n}{l} q_x^l (1-q_x)^{n-l}$, where $q_x = p_x/2$ is the probability that at step 6.3 A puts the element z in the stack. Then the probability of putting j triples in the stack, for each clause c_i, is

$$\binom{n}{3j} q_x^{3j}(1-q_x)^{n-3j} + \binom{n}{3j+1} q_x^{3j+1}(1-q_x)^{n-3j-1} + \binom{n}{3j+2} q_x^{3j+2}(1-q_x)^{n-3j-2}.$$

Therefore

$$\begin{aligned} Pr(R) &\leq \left(\frac{8}{7}\right)^{2/3} (1-q_x)^n \left(1 + \frac{\sqrt[3]{7}}{2} \frac{q_x}{1-q_x}\right)^{3\lfloor n/3 \rfloor + 2} \\ &= (1-q_x)^{n-3\lfloor n/3 \rfloor - 2} \left(\frac{8}{7}\right)^{2/3} \left(1 + \left(\frac{\sqrt[3]{7}}{2} - 1\right) q_x\right)^{3\lfloor n/3 \rfloor + 2} \\ &= O\big((1 - .02 p_x)^n\big). \end{aligned}$$

(A,B) satisfies the Zero-Knowledge requirement.

For the Zero-Knowledge part, we exhibit, a Probabilistic Turing machine M that, in expected polynomial time, approximates the family of random variables $V = \{V(x)\}$, where $V(x) = \{\sigma \leftarrow \{0,1\}^{|x|^c}; y \leftarrow A(\sigma,x): (\sigma,y)\}$, over $3SAT$.

M's simulation protocol.

1) M chooses a random $x \in Z_2(n)$ along with its factorization, $x = pq$.
2) M sets $\sigma =$ empty string and repeats step 3 $7n$ times.
3) M chooses a random n-bit integer ν. If either $\nu \notin Z_x^{+1}$ then M adds the binary representation of ν to σ. Otherwise M chooses a random ρ in Z_x^* and adds the binary representation of $\rho^2 \bmod x$ to σ.
4) M adds n^3 random bits to σ. Let $\sigma = \sigma_0 \ldots \sigma_{7n^2+n^3-1}$.
5) M sets $E_0, E_1 =$ empty set.
6) For $i = 1, \ldots, 7n$, let s_i be the integer whose binary representation is $\sigma_{(i-1)n} \cdots \sigma_{in-1}$. If $s_i \notin Z_x^{+1}$ then M discards it. Otherwise M tosses a fair coin. If HEAD (TAIL) then M puts s_i in $E_0(E_1)$. If $E_0(E_1)$ is not empty M writes $(0,s,\alpha)\Big((1,s,\alpha)\Big)$, where s is a random element in $E_0(E_1)$ and α is a random square root modulo x of ss_i. If $E_0(E_1)$ is empty M writes $(0,0,\alpha)\Big((1,0,\alpha)\Big)$.
7) If E_0 or E_1 (or both) is empty then M halts.
   Otherwise, M randomly chooses $y_0 \in E_0$ and $y_1 \in E_1$, computes $y = y_0 y_1 \bmod x$ and writes (y, y_0, y_1).
8) M picks at random $r_j \in Z_x^*$, $j = 1, \ldots, k$, computes $w_j = r_j^2 \bmod x$ and writes $(w_1, w_2, \ldots, w_k)$.
9) From this point on, M performs the same protocol as A.
   (Note that M can perform A's protocol in polynomial time since he knows the factorization of x.)

The output of M is different from that of A only because in one case the string σ is random while in the other its first $7n^2$ bits are either the binary representation of elements not in Z_x^{+1} or quadratic residues modulo x. Under the quadratic residuosity assumption these two distributions are indistinguishable (which is not hard to prove).

3.2 A stronger version of our result.

The Single-Theorem Non-Interactive ZKPS of section 3.1 has a limited applicability. This is a drawback that is best illustrated by our conceptual example of the prover A who is leaving for his trip.

It is unlikely that for each theorem $\mathcal{T}$ that A finds, a string $\sigma_{\mathcal{T}}$ comes from the sky "devoted" to $\mathcal{T}$ and is presented to (is read by) both A and B. It is instead more probable, that A and B may have witnessed the same common random event of size n once, when and because they were together (or else, it is more probable that they generated a (common) random event. For instance by the coin flipping protocol as explained in section 1.1).

However the Proof System of section 3.1 will enable A to subsequently prove in Zero-Knowledge to B only a theorem of smaller size, roughly $\sqrt[3]{n}$ bit long. He is out of luck would he discover the proof of a theorem of bigger size.

Moreover, the n-bit long string A leaves with will not enable him to not interactively and in Zero-Knowledge prove to B many theorems. Below we modify the definition of Non-Interactive Zero-Knowledge Proof Systems with common coins and our solution so to allow A to prove to a B with which he shares an n-bit string, $poly(n)$ theorems of $poly(n)$ size.

We first define formally what this means.

Definition. *A Non-Interactive ZKPS is a pair (A,B) where A is a pair, (A_0, A_1), of Probabilistic Turing Machines and $B(\cdot,\cdot,\cdot,\cdot)$ is a deterministic algorithm running in time polynomial in the length of its first input, such that:*

1) **(Completeness)** *For all polynomials P, Q, and for all $(x_1, x_2, \ldots, x_{Q(n)}) \in (3SAT_{P(n)})^{Q(n)}$*

$$Pr\Big(\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_0 \leftarrow A_0(\sigma);\ y_1 \leftarrow A_1(\sigma, x_1, y_0);\ \vdots \quad \vdots \quad y_{Q(n)} \leftarrow A_1(\sigma, x_{Q(n)}, y_0) : \bigwedge_{j=1}^{Q(n)} B(x_j, y_j, y_0, \sigma) = 1\Big) > 1 - n^{-O(1)}.$$

2) **(Soundness)** *For all polynomials P, Q, for all $(x_1, x_2, \ldots, x_{Q(n)}) \notin (3SAT_{P(n)})^{Q(n)}$*

and for each $A' = (A'_0, A'_1)$

$$Pr\Big(\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_0 \leftarrow A'_0(\sigma);\ y_1 \leftarrow A'_1(\sigma, x_1, y_0);\ \vdots \quad \vdots\ ;\ y_{Q(n)} \leftarrow A'_1(\sigma, x_{Q(n)}, y_0) : \bigwedge_{j=1}^{Q(n)} B(x_j, y_j, y_0, \sigma) = 1\Big) < n^{-O(1)}.$$

3) **(Zero-Knowledge)** *For each polynomial* Q*, the family of random variables* $V = \{V(x_1, \ldots, x_{Q(n)})\}$*, where*

$$V(x_1, \ldots, x_{Q(n)}) = \Big\{\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_0 \leftarrow A_0(\sigma);\ y_1 \leftarrow A_1(\sigma, x_1, y_0);\ \vdots \quad \vdots\ ;\ y_{Q(n)} \leftarrow A_1(\sigma, x_{Q(n)}, y_0)\ : (\sigma, y_0, y_1, \ldots, y_{Q(n)})\Big\}$$

is approximable over $\bigcup_n (3SAT)^{Q(n)}$.

Comment. Notice that we let A send first the proof of a theorem (y_0) other than the ones he is interested in communicating: $x_1, x_2, \ldots$. Notice that A is able to prove each "interesting" theorem essentially independently from all the other theorems (only memory of y_0 is needed by A to prove each single theorem). Also A need only to be poly-time if he has a suitable witness among his available inputs. This is not inherent to our definition of Non-Interactive ZKPS's. It is rather due to our specific solution. However it does not change the rules of the game in an essential way.

Notice also that B verifies each theorem independently (only memory of y_0 is needed); we essentially have one prover and many independent verifiers. Moreover, the proof of each subset of the interesting theorems is Zero-Knowledge.

Theorem 2. *Under the QRA, there exists a Non-Interactive ZKPS.*

Proof. The proof is similar to that of Theorem 1 but more technical and cumbersome. Our protocol uses random functions [GGM] to "expand" the initial random seed. It will appear in the final paper. □

3.3 Open Problems.

Our results can be extended in two directions.

The first extension deals with a scenario in which we have many independent provers, using the same random string σ to prove different theorems. A partial solution of this problem will appear in the final paper.

The second extension concerns the Complexity Theoretic Assumption on which our results are based. Namely, can we replace the QRA with the weaker assumption of the Existence of One-way functions? This question is discussed in the following and we address the reader to [DMP] for a partial solution of it.

3.3.1 Many Independent Provers.

We live in a scientific community in which all libraries possess copies of the same tables of random numbers prepeared by RAND corporation, the RAND tables. This is essentially a short string *shared* by the scientific community. Can we use the RAND tables to give one another Non-Interactive Zero-Knowledge Proofs?

Here the problem is not so much the fact that we share a random string of fixed length, rather than σ_n for each n. In fact the RAND table is long enough to allow us to prove an arbitrary polynomial number of theorems. The fact is that 3.2 tells us that a *single* prover releases Zero-Knowledge. Is this also true if we have (as it is the case) many provers? This problem is similar to that discussed in section 3.2.

We know that the answer is "Yes" if at most $O(\log n)$ provers are active, when a string of length n is available. The protocol can be accommodated to $M(n)$ provers. However each prover is obliged to invest a multiplicative factor of $M(n)$ in his computational effort (whether or not there really are $M(n)$ provers). This is unsatisfactory. It should be contrasted with the $P(\cdot)$ size and with the $Q(\cdot)$ many theorems of protocol 3.2.

We are thus naturally led to the definition of *Economical Non-Interactive ZKPS.*

Definition. *An Economical Non-Interactive ZKPS is a pair (A,B) where A is a Probabilistic Turing Machine and $B(\cdot,\cdot,\cdot)$ is a deterministic algorithm running in time polynomial in the length of its first input, such that:*

1) **(Completeness)** *For all polynomials P, Q, and for all $(x_1, x_2, \ldots, x_{Q(n)}) \in$*

$(3SAT_{P(n)})^{Q(n)}$

$$Pr\Big(\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_1 \leftarrow A(\sigma, x_1);$$

$$\vdots \qquad\qquad \vdots$$

$$y_{Q(n)} \leftarrow A(\sigma, x_{Q(n)}) \ :\ \bigwedge_{j=1}^{Q(n)} B(x_j, y_j, \sigma) = 1\Big) > 1 - n^{-O(1)}.$$

2) **(Soundness)** *For all polynomials P, Q, for all $(x_1, x_2, \ldots, x_{Q(n)}) \notin (3SAT_{P(n)})^{Q(n)}$ and for each A'*

$$Pr\Big(\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_1 \leftarrow A'(\sigma, x_1);$$

$$\vdots \qquad\qquad \vdots$$

$$y_{Q(n)} \leftarrow A'(\sigma, x_{Q(n)}, y_0) : \bigwedge_{j=1}^{Q(n)} B(x_j, y_j, \sigma) = 1\Big) < n^{-O(1)}.$$

3) **(Zero-Knowledge)** *For each polynomial Q, the family of random variables $V = \{V(x_1, \ldots, x_{Q(n)})\}$, where*

$$V(x_1, \ldots, x_{Q(n)}) = \Big\{\sigma \leftarrow \{0,1\}^{n^{O(1)}};\ y_1 \leftarrow A(\sigma, x_1);$$

$$\vdots \qquad\qquad \vdots$$

$$y_{Q(n)} \leftarrow A(\sigma, x_{Q(n)}) : (\sigma, y_1, \ldots, y_{Q(n)})\Big\}$$

is approximable over $\bigcup_n (3SAT)^{Q(n)}$.

Comment. Notice that the above definition is different from that of Non-Interactive ZKPS in the requirement that a proof of a theorem depends only on σ and not on any previously proved theorem (y_0).

3.3.2 Relaxing the assumption.

Our protocol relies on the fact that deciding Quadratic Residuosity is *hard.*

One would like to prove our result under the assumption that one-way functions exist. This is the weakest possible assumption in Cryptography, since if one-way functions do not exist then public-key cryptography is not possible. In [DMP] we present a partial solution

to this problem. Namely we exhibit a protocol that allows a prover to non interactively prove any theorem of size n after an interactive preprocessing step whose computational effort is roughly n^3.

References.

[B] M. Blum, *Coin Flipping By Telephone*, IEEE COMPCON '82.

[BC] G. Brassard, C. Crepeau, *Non Transitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for Sat and Beyond*, Proceedings of the 27th Symposium on Foundations of Computer Science, 1986.

[BFM] M. Blum, P. Feldman, S. Micali, in preparation.

[DMP] A. De Santis, S. Micali, G. Persiano, in preparation.

[FMRW] M. Fischer, S. Micali, C. Rackoff and D. Witenberg, *A Secure Protocol for the Oblivious Transfer*, in preparation 1986.

[FFS] U. Feige, A. Fiat and A. Shamir, *Zero Knowledge Proofs of Identity*, Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987.

[GGM] O. Goldreich, S. Goldwasser, S. Micali, *How to Construct Random Functions*, Journal of ACM, vol. 33, No. 4, October 1986.

[GHY] Z. Galil, S. Haber, M. Yung, *A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge Public-Key Cryptosystem*, Proceedings of the 26th Symposium on Foundations of Computer-Science, 1985.

[GJ] M. Garey, D. Johnson, *Computers and Intractability : a Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, 1979.

[GM] S. Goldwasser, S. Micali, *Probabilistic Encryption*, Journal of Computer and System Science, vol. 28, No. 2, 1984.

[GMR] S. Goldwasser, S. Micali, C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, Proceedings of the 17th Annual ACM Symposium on Theory of Computing, 1985.

[GMW1] O. Goldreich, S. Micali, A. Wigderson, *Proofs That Yield Nothing But Their Validity and a Methodology of Cryptographic Protocols Design,* Proceedings of the 27th Symposium on Foundations of Computer-Science, 1986.

[GMW2] O. Goldreich, S. Micali, A. Wigderson, *How to Play Any Mental Game,* Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987.

[GoMiRi] S. Goldwasser, S. Micali, R. Rivest, *A Digital Signature Scheme Secure Against Adaptive, Chosen Cyphertext Attack,* to appear in SIAM J. on Computing.