Cryptographic Applications of the Non-Interactive Metaproof and Many-prover Systems

Alfredo De Santis*
Dipartimento di Informatica ed Applicazioni
Università di Salerno
84081 Baronissi (Salerno), Italy

Moti Yung
IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

Preliminary Version

Abstract

In a companion paper [DeYu] we have developed the tool of non-interactive proof-system we call "Metaproof" (*μ-NIZK proof system*); this provides a proof of "the existence of a proof to a statement". Using a reduction of the theorem to a set of claims about encrypted values, enabled us to develop a crucial proof-system property which we called *"on-line simulatable NIZK proof-system"*. This was used to implement the "Many-Prover Non-Interactive Proof-System" where independent users can send proofs (which was not known in the original system and was open), and a "Self-Referential NIZK proof system" where the random reference string is available to the polynomial-time opponent who chooses the theorem to prove, (this was an intriguing question regarding such systems).

In this abstract we present an introduction to the basic tools and their possible applications. The subject of this paper is a variety of cryptographic applications provided by the new tools. We demonstrate its applicability in enhancing security and properties of a methodology for signature and authentication developed by Bellare and Goldwasser [BeGo] (by using the Metaproof system to solve the open problem of many-prover NIZK system). We also show, among other things, how the tools can be used to provide security mechanisms such as an "Oblivious Warden" which translates non-interactive proofs to random ones independently of the proof itself, and the notion of "Gradual opening of a zero-knowledge computation" which is first demonstrated to be correct using a non-interactive proof, and then is opened gradually and fast (i.e., without further proofs).

*Part of this work was done while the author was visiting IBM Research Division, T. J. Watson Research Ctr, Yorktown Heights, NY 10598.

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 366-377, 1991.

1 Introduction

The development of zero-knowledge proof-systems introduced by Goldwasser, Micali, and Rackoff [GoMiRa] has revolutionized the field of cryptographic primitives and protocols design. Designing of basic primitives was made possible and proving security was made easy.

Two very useful notions of zero-knowledge proofs were given: interactive proofs (ZKIP) [GoMiRa, GoMiWi1, ImYu] and non-interactive proofs (NIZK) (introduced by Blum, Feldman, and Micali) [BlFeMi, DeMiPe1, BlDeMiPe]. Next we evaluate the current relative advantages and shortcomings of the two notions.

In the non-interactive model indeed interaction was shown not to be a necessary ingredient of zero-knowledge proofs. It was replaced by a shared public (short) string of random bits. The motivation for such a model is the availability of public random sources. For instance, a community in which everyone possess (in the local library) a copy of the same tables of random numbers prepared by RAND corporation, the RAND tables. NIZK is a new tool which is not well-understood yet and many more questions regarding the possibility of NIZK proof-systems are intriguing questions, we have recently solved some of them [DeYu]. In this work we present applications of the new tools.

Our results:
In [DeYu], we have introduced two new non-interactive tools. The first one is a "Metaproof system". A meta-proof in terms of proof-theory is *a proof that there is a proof for a theorem.* Indeed the development of deductive systems and the notion of system which proves theorem and then another logical metasystem which proves theorems about theorems, is one of the major development of foundations of mathematics in the last century. However, for a language in $\mathcal{NP}$, in the usual sense a proof of an existence of a proof is just another way (possibly a weird one) to claim that the original theorem is in the language by claiming that there is a proof for it. Nevertheless, in the context of the theory of zero-knowledge proofs, we will show that this way of proving the existence of a proof is a very useful tool. One implication of the tool is an implementation of the Many-prover NIZK (MP-NIZK), an extension of NIZK to many users, which was an open question.

This work presents a variety of applications of μ-NIZK and MP-NIZK in typical cryptographic settings. We present applications to encryption and signature schemes, and numerous cryptographic protocols and other primitives.

Among our applications are extensions and improvements of the very nice set of cryptographic applications suggested by the paradigm of Bellare and Goldwasser [BeGo], who showed how combining pseudo-random functions, encryptions, and zero-knowledge proofs gives signature and similar primitives. We show how to implement such notions as history-independent signature schemes, and ID distribution in the context of identification schemes, with security as high as the encryption functions and the non-interactive zero-knowledge proofs involved.

Another application is in the domain of resource abuse prevention. An *Oblivious Warden* whose task is to eliminate abuses of a channel [De]. Our tool eliminates additional (illegal) information embedded in a messages which is sent as non-interactive proofs. The warden changes the message while maintaining the semantics of the proof.

Meta-proofs can be nested. We note that the Meta-proof sequencing and nesting property shows that the metaproof is more than just an indirect proof of a theorem, a property which helps in achieving the MP-NIZK (in this sense this is similar to [FeLaSh]). However, the Meta-proof notion is also a tool for combining proofs and enables the forwarding of a proof based on a zero-knowledge witness rather than the "real" witness itself which gives the flexibility which is the strength behind some of our applications. In this paper we mainly present these applications.

2 Preliminaries

Next, we present the necessary background: the basic definitions, bounded NIZK, and encryption functions.

2.1 Basic definitions.

A sequence of probabilistic Turing machines $\{T_n\}_{n \in \mathcal{N}}$ is an *efficient non-uniform algorithm* if there exists a positive constant c such that, for all sufficiently large n, T_n halts in expected n^c and the size of its program is $\leq n^c$. We use efficient non-uniform algorithms to gain the power of using different Turing machines for different input lengths. For instance, T_n can be used for inputs of length n. The power of non-uniformity lies in the fact that each Turing machine in the sequence may have "wired-in" (i.e. properly encoded in its program) a small amount of special information about its own input length.

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. Notice that we do not mention explicitly the random coins used by probabilistic algorithms.

If $p(\cdot,\cdot,\cdots)$ is a predicate, the notation $Pr(x \overset{R}{\leftarrow} S; y \overset{R}{\leftarrow} T; ... : p(x,y,\cdots))$ denotes the probability that $p(x,y,\cdots)$ will be true after the ordered execution of the algorithms $x \overset{R}{\leftarrow} S$, $y \overset{R}{\leftarrow} T$, ...

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

A notion which is important in our context is *"a history-insensitive algorithm"*. A probabilistic Turing machine R has *one-way input tape*, *one-way output tape*, *one-way random tape*, and a regular *work tape*. A one-way tape is a tape in which after each read/write operation the head moves, always from left to right. R is called a history-insensitive algorithm if it works as follows. First, after copying the input to its work tape, R produces the output and writes it on the output tape. Then, R erases its work tape and returns to its initial state, without backtracking the one-way heads.

2.2 Bounded Non-Interactive Zero-knowledge Proof Systems

Bounded NIZK proof systems were conceived by Blum, Feldman, and Micali, and were presented in [BlFeMi], [DeMiPe1], and [BlDeMiPe]. The term "Bounded" refers to the fact that the proof system is defined for a single theorem or a few "short" theorems. Without loss of generality we use a complete language in $\mathcal{NP}$: $3SAT$ [Co].

Definition 2.1 Let A_1 and A_2 be Turing Machines. We say that (A_1, A_2) is a *sender–receiver* pair if their computation on a *common input* x works as follows. First, algorithm A_1, on input x, outputs a string (message) m_x. Then, algorithm A_2, computes on inputs x and m_x and outputs ACCEPT or REJECT. A_1 is called the sender and A_2 the receiver. The running times of both machines is calculated in terms of the common input.

Definition 2.2 Let (*Prover*, *Verifier*) be a sender–receiver pair, where *Prover* is history-insensitive and *Verifier* is polynomial-time. We say that (*Prover*, *Verifier*), is a Bounded Non-Interactive Zero-Knowledge Proof System (Bounded NIZK proof system) for $3SAT$ if there exists a positive constant c such that:

1. *Completeness.* $\forall \Phi \in 3SAT$ and satisfying assignments t,

$$Pr(\sigma \overset{R}{\leftarrow} \{0,1\}^{n^c}; Proof \overset{R}{\leftarrow} Prover(\sigma, \Phi, t) : Verifier(\sigma. \Phi. Proof) = 1) = 1.$$

2. *Soundness.* For all probabilistic algorithms *Adversary* outputting pairs $(\Phi, Proof)$, where $\Phi \notin 3SAT_n$, $\forall d > 0$, and all sufficiently large n,

$$Pr(\sigma \overset{R}{\leftarrow} \{0,1\}^{n^c}; (\Phi, Proof) \overset{R}{\leftarrow} Adversary(\sigma) : Verifier(\sigma. \Phi, Proof) = 1) < n^{-d}.$$

3. *Zero-Knowledge.* There exists an efficient algorithm S such that $\forall \Phi \in 3SAT_n$, for all satisfying assignments t for Φ, for all efficient non-uniform (distinguishing) algorithms D, $\forall d > 0$, and all sufficiently large n,

$$\left| Pr(s \overset{R}{\leftarrow} View(n, \Phi, t) : D_n(s) = 1) - Pr(s \overset{R}{\leftarrow} S(1^n, \Phi) : D_n(s) = 1) \right| < n^{-d},$$

where

$$View(n, \Phi, t) = \{\sigma \overset{R}{\leftarrow} \{0,1\}^{n^c}; Proof \overset{R}{\leftarrow} Prover(\sigma, \Phi, t) : (\sigma, Proof)\}.$$

We call algorithm S the *Simulator*.

A sender-receiver pair (*Prover, Verifier*) is a Bounded Non-Interactive Proof System for $3SAT$ if there exists a positive constant c such that completeness and soundness hold (such a c will be referred as the *constant* of (*Prover, Verifier*)).
We call the "common" random string σ, input to both *Prover* and *Verifier*, the *reference string.* (Above σ and Φ are the common input.)
In the above definition, there is no limitation on the running time of *Prover.* In cryptographic applications it is required that the prover be expected polynomial time. A Bounded NIZK proof system (*Prover, Verifier*) with an efficient prover is a Bounded NIZK proof system where *Prover* on any common input (i.e. a reference string and a formula) runs in expected polynomial time. Recently, it was shown how to base bounded non-interactive system on any one-way permutation [LaSh, FeLaSh].

2.3 Non-Interactive Encryption Tools

We have developed two encryption tools. First, we have shown that *secure probabilistic encryption scheme* exists in the non-interactive model, based on any one way function. (We use Naor's bit-commitment protocol [Na]). Next, an *ambiguous encryption* is possible in the model, based on a one-way function; it is a good bit encryption method in the model which can be simulated to be opened as both bits!

3 Review: New Non-Interactive Proof Systems

We next describe the recent non-interactive systems we have designed.

3.1 Metaproofs

Roughly speaking, the "metaproof of a theorem T" is a NIZK proof that "there is a NIZK proof of T". Assume that, for a formula Φ, there is a NIZK proof *pf* computed using a reference string σ_1. The *metaprover* μP on input a formula Φ and *pf*, computes a NIZK proof *Metaproof* using a different reference string σ_2, that there is indeed a string *pf* such that *Verifier* would accept as proof of Φ. The *metaverifier* μV checks that *Metaproof* has been correctly computed, but has no access whatsoever to *pf*.

More formally, let (*Prover*, *Verifier*) be a Bounded NIZK proof system for $3SAT$. Next, let $Prover(\sigma, \Phi, t)$ be an efficient *Prover*'s program that uses σ as its reference string and the satisfying assignment t to prove $\Phi \in 3SAT_n$. Thus, there is a constant $c > 0$ such that on input $r \in \{0,1\}^{n^c}$, $\Phi \in 3SAT_n$, and a satisfying assignment t, *Prover* computes a string *pf*, $|pf| \leq n^c$, such that $Verifier(r, \Phi, pf) = 1$. Let $L = \bigcup_n L(n)$ be the language where

$$L(n) = \{(r, \Phi) : |r| = n^c,\ \Phi \in 3SAT_n, \text{ and } \exists pf, |pf| \leq n^c \text{ such that } Verifier(r, \Phi, pf) = 1\}.$$

Then $\Phi \in 3SAT_n$ iff $(r, \Phi) \in L(n)$ for all strings r. Moreover $L \in \mathcal{NP}$ and thus there is a fixed polynomial-time computable reduction $REDUCE$ such that

$$(r, \Phi) \in L(n) \iff \Psi = REDUCE(r, \Phi) \in 3SAT_{n^b}$$

where $b > 0$ is a fixed constant depending only on the reduction $REDUCE$. More precisely, the formula Ψ is obtained by encoding the computation of *Verifier* on input r, Φ, pf as in Cook's Theorem, and then reducing it to a 3-satisfiable formula, as in [Co]. A well known property of this reduction is that to each "witness" *pf* one can associate in polynomial-time a satisfying assignment α for Ψ. We call $Witness(r, \Phi, pf)$ the poly-time procedure that returns the satisfying assignment α for $\Psi = REDUCE(r, \Phi)$.

Now, we describe the programs for the metaprover $\mu P(\cdot, \cdot, \cdot)$ and the metaverifier $\mu V(\cdot, \cdot)$

The sender–receiver pair $(\mu P, \mu V)$

Input to μP and μV:

- A random string $\sigma_1 \circ \sigma_2$, where $|\sigma_1| = n^c$ and $|\sigma_2| = n^{bc}$.
- $\Phi \in 3SAT_n$.

Instructions for μP

Private Input: a string *pf* such that $Verifier(\sigma_1, \Phi, pf) = 1$.

$\mu P.1$ Compute $\Psi = REDUCE(\sigma_1, \Phi)$ and $\alpha = Witness(\sigma_1, \Phi, pf)$.

$\mu P.2$ Run $Prover(\sigma_2, \Psi, \alpha)$. Call *Metaproof* the output and send it to μV.

```
Instructions for $\mu V$
Input from $\mu P$: a string $Metaproof$.
$\mu V.0$ Compute $n$ from $\sigma_1 \circ \sigma_2$.
    Verify that $\Phi$ has at most $n$ clauses with 3 literals each. If not, REJECT.
$\mu V.1$ Compute the formula $\Psi = REDUCE(\sigma_1, \Phi)$.
$\mu V.2$ If $Verifier(\sigma_2, \Psi, Metaproof) = 1$ then ACCEPT. Else, REJECT.
```

We formally prove in [DeYu] that the metaproof system $(\mu P, \mu V)$ above is a Bounded NIZK proof system for $3SAT$. Indeed, the tool does not seem to help, especially since the meta-prover need not be more than an efficient program. However, this intuition is wrong.

3.2 Theorem Translation and On-line Simulation

The original notion of Bounded-NIZK was defined to be zero-knowledge by exhibiting a simulator which generates transcripts of reference strings and proofs which a polynomial machine cannot tell apart from real proofs. The simulator was defined as a machine which first gets the theorem to be proved and then starts the computation. Next we define and implement a simulator which works in a different mode. In a preprocessing stage the processor prepares a prefix of a simulation (the reference string); when given a theorem, the simulated proof with respect to the reference string is generated.

This fashion of simulation resembles ideas presented in simulations in [DeMiPe2, ImYu]. It is not clear that it is a stronger definition, however, this simulation mode will be instrumental in constructing a many-provers NIZK proof system using metaproofs.

Definition 3.1 Let $(Prover, Verifier)$ be a Bounded NIZK proof system for $3SAT$. A simulator $M(\cdot,\cdot)$ for $(Prover, Verifier)$ is an *on-line simulator* if it consists of a pair of efficient algorithms $M = (M_1, M_2)$ that work as follows: First it gets as input 1^n and it compute: $(\sigma, state) \overset{R}{\leftarrow} M_1(1^n)$. Then it gets as second input: $x \in 3SAT_n$ and it computes $Proof \overset{R}{\leftarrow} M_2(state, x)$. It outputs: $(\sigma, Proof)$.

A Bounded NIZK proof system for $3SAT$ is *on-line simulatable* if it has an on-line simulator.

Theorem Translation: an overview of the construction.

The idea is to prepare a machinery for the proof based on ciphertexts of an encryption function in an off line fashion, independently of the proof. That is, the statement of the proof is reduced to claims about certain encrypted values. In the simulation the public string can therefore be prepared independently of the theorem to be proved. This enables the on-line simulatable proof-system as the public string is independent of the proof itself. The idea can be viewed as a bootstrapping method on the usual method of hiding part of the proof using encryption which is usually done in ZKIP and NIZK systems which rely on encryption (e.g., [GoMiWil]). More details are given in [DeYu]

3.3 Many-provers Non-Interactive Zero-knowledge Proof Systems

Consider a scenario in which we have many independent provers, using the same random string σ to prove different theorems. For instance, a scientific community in which all libraries possess copies of the same tables of random numbers prepared by RAND corporation, the RAND tables. This is essentially a short string *shared* by the scientific community. Can they use the RAND tables to give one another Non-Interactive Zero-Knowledge Proofs? (see [DeMiPel]) A many-provers NIZK proof system is a solution to this problem.

Definition 3.2 Let (*Prover*, *Verifier*) be a sender-receiver pair, where *Prover* is history-insensitive and *Verifier* is polynomial time. We say that (*Prover*, *Verifier*) is a Many-provers Non-Interactive Zero-Knowledge Proof System (MP-NIZK proof system) if the following 3 conditions hold.

1. *Completeness.* $\forall\Phi \in 3SAT$, and for all satisfying assignments t for Φ,

$$Pr\Big(\sigma \overset{R}{\leftarrow} \{0,1\}^n; Proof \overset{R}{\leftarrow} Prover(\sigma,\Phi,t) : Verifier(\sigma,\Phi,Proof) = 1\Big) = 1.$$

2. *Soundness.* For all probabilistic algorithms *Adversary* outputting pairs $(\Phi', Proof')$, where $\Phi' \notin 3SAT$, $\forall d > 0$, and all sufficiently large n,

$$Pr\Big(\sigma \overset{R}{\leftarrow} \{0,1\}^n; (\Phi', Proof') \overset{R}{\leftarrow} Adversary(\sigma) : Verifier\,(\sigma,\Phi',Proof') = 1\Big) < n^{-d}.$$

3. *Zero-Knowledge.* There exists an efficient algorithm S such that $\forall\Phi_1,\Phi_2,\ldots \in 3SAT$, for all satisfying assignments $t_1, t_2, \ldots$, for all efficient non-uniform algorithms D, $\forall d > 0$, and all sufficiently large n,

$$\Big|Pr(s \overset{R}{\leftarrow} View(n,\Phi_1,t_1,\Phi_2,t_2,\ldots) : D_n(s) = 1) - Pr(s \overset{R}{\leftarrow} S(1^n,\Phi_1,\Phi_2,\ldots) : D_n(s) = 1)\Big| < n^{-d}$$

where

$$\begin{aligned} View(n,\Phi_1,t_1,\Phi_2,t_2,\ldots) = \Big\{\sigma \overset{R}{\leftarrow} \{0,1\}^n;\ & Proof_1 \overset{R}{\leftarrow} Prover(\sigma,\Phi_1,t_1); \\ & Proof_2 \overset{R}{\leftarrow} Prover(\sigma,\Phi_2,t_2); \\ & \quad\vdots \\ & : (\sigma, Proof_1, Proof_2, \ldots)\Big\}. \end{aligned}$$

We call *Simulator* the algorithm S.

A sender-receiver pair (*Prover*, *Verifier*) is a Non-Interactive Proof System for $3SAT$ if Completeness and Soundness hold. An alternative definition of the zero-knowledge property is that there are several independent provers, each using the same algorithm and the same reference string, but its own private random string. Since the prover is history-insensitive these two definitions are equivalent. When the metaproofs are combined with an on-line simulatable bounded NIZK proof system, they give a protocol for many-provers NIZK proof systems.

In [DeYu], we describe a sender-receiver pair (P, V). P can prove in zero-knowledge the 3-satisfiability of any number of 3-satisfiable formulae with n clauses each. We then employ a technique of [BlDeMiPe] to extend this by showing how to use the same protocol to prove any number of formulae, each of arbitrary size.

3.4 Self-Referential NIZK

Another important intriguing problem is that the non-interactive model is shown zero-knowledge based on random reference string which is available on-line, and not off-line, in advance. On the other hand the motivation for such a tool is the availability of random public sources "the Rand books". This has been bothering researchers and, for example, Bellare and Goldwasser [BeGo] presented a definition of a strong non-interactive system which allows on-line randomness (maybe after a stage of preprocessing). They did not have (and actually did not need) such a system.

Based on the meta-proof system we can finally have an "on-line" system, where the polynomial-time theorem chooser is getting access to the random reference string. (The theorem can rely on this string and thus the system can be self-referential in the sense that the same string will be used to prove the correctness of the theorem). This increases the applicability of the NIZK systems to many more scenarios and protocols.

4 Applications to Identification and Signature

The goal of this paper is to show how to use the above notions. The primitives above can be applied by efficient (polynomial-time) users of a cryptographic system and thus can be applied in cryptographic settings. They give a variety of applications and new tools for secure systems. The applications use a few facts. First, the fact that the metaproof system gives a proof in an indirect fashion, covered by additional encryption mechanism. Second, the in the metaproof the metaprover possesses a zero-knowledge witness and does not have to have a real knowledge about the witness of the proof itself. Third fact, is that metaproof can be applied recursively.

We start with applications related to a new methodology suggested recently by Bellare and Goldwasser.

4.1 Signatures without history

Based on secure probabilistic encryption schemes and NIZK, Bellare and Goldwasser [BeGo] have suggested a signature scheme which relies on the two tools. More specifically, they use a NIZK proof systems which is publicly verifiable and a pseudo-random functions collection. Their method uses an encryption of a seed of a pseudorandom function, and uses non-interactive proofs to show that certain activity related to the message and the encryption is performed correctly; only the signer is able to perform the task. (For more details see [BeGo]). They construct a scheme which is secure against a random message attack.

Even, Goldreich, and Micali [EvGoMi] proved that any scheme secure against random message attack (and memoryless) can be transformed into one secure against adaptive chosen message attack (and memoryless).

The scheme using NIZK is not memory-less since for the original NIZK, a proof depends on previously given proofs. To prevent history-dependence Bellare and Goldwasser suggest a NIZK which involves initial preprocessing or reliance on a trusted center. Then, Feige and Shamir [FeSh] by relaxing the security requirement to Witness-Hiding rather than Zero-Knowledge can apply the protocol without initial interaction.

By using MP-NIZK rather than NIZK (which is history dependent) we finally achieve a signature system secure against adaptive chosen plaintext attack in the paradigm of [BeGo], which is history-free, preprocessing-free (no trusted server as well), and *without* relaxation of the original security definition.

The signature schemes based on Universal One-way hash functions (UOWHF), an approach initiated in [NaYu] which proved that a trapdoor-less provably-secure signature is possible, has all its implementations history-dependent.

4.2 Hierarchical identification.

Another application mentioned in [BeGo] is identification schemes. We extend the notion (using the metaproof system) to enable hierarchical distribution of identification information. The original system enables a center to distribute unforgeable ID numbers. With the metaproof system we can implement a hierarchical system in which a center can issue ID's to sub-centers (officers), later the local center can transfer the ID's on. Based on metaproofs level (metaproof, metametaproof, etc.), the user can verify the authenticity of ID's and the level of the officer which is giving the ID. This hierarchical center structure is typical to large organizations.

5 Applications to Encryption, Non-Interactive Proofs, and Resource Protection

5.1 Enhancing security by sequencing proofs

An immediate application of the Metaproof is cascading a constant number of metaproof systems to enhance security of zero-knowledge schemes. Using a, metaproof, one can hide the previous proof applying a different encryption key (we view the proofs as nested). In each level one can use a key and even if only one of the keys is secure, the entire proof is secure (zero-knowledge). It is important to notice that even if certain outer-most levels are insecure, the system is still secure. The outer-most secure claim has the zero-knowledge property and when the opponent decrypts the claims he obtains a claim which is actually encrypted securely.

The level of a meta-proof may be used to trace the proof (given with a signature) back to its origin if a proof is passed among users.

5.2 Abuse-freeness: the oblivious warden

Desmedt [De] has introduced the notion of protecting against the abuse of channels used for cryptographic tasks. This is a classical problem of prevention of abuse of resources and prevention of violations by users. His example is an authentication channel whose users do not care about the authentication, but try to convey extra information in the process of authenticating themselves. Desmedt suggests to protect channels by assigning wardens to monitor and modify information passing through the channel. He gives a nice set of techniques which enable abuse-freeness of protocols. He also suggests to use NIZK by which the sender proves to the warden that it follows the protocol correctly with respect to some initial interactive commitment.

Using the notion of the metaproof, a different idea can be developed. A NIZK system used by the sender can be made abuse-free by having an oblivious warden (whose task is to prevent violations). The warden simply gets a proof of a theorem, verifies it and forward to the receiver the metaproof, rather then the original proof. He is able to convince the receiver just as well as the original sender, but by sending a proof which is based on his own random bits. The warden's task is independent of the actual proof or NIZK system in use so it can be a general procedure which obliviously translates NIZK proofs and thus assures they are abuse-free.

5.3 Gradual NIZK result-opening

In [ImYu] it was shown how an adaptive verifier can open a result of a computation which was proven to him (encrypted computation performed in zero-knowledge) by the prover. The opening of the result is requested bit by bit by the verifier, and the prover opens the result (fast— without further proofs- for efficiency reasons), each time the verifier can decide which question to ask next. This implies that the simulator in the zero-knowledge proof (which does not know the result but gets it bit by bit as well) should also be able to open result bits of its proof. The difficulty is that the simulator has to open the result bits in an on-line fashion, (after he committed to the proof of the computing circuit). The simulator, on the other hand, should not use more knowledge than the on-line already opened bits and the current result to be opened (which it gets from a "result oracle"). How can this be done without performing the opening stages themselves in a zero-knowledge fashion (which requires further proofs!)?

This can be achieved in the NIZK scenario as well, using the tool of ambiguous non-interactive encryption which we developed, thus closing another gap between the interactive and the non-interactive scenarios.

5.4 Combination Transferable/Non-Transferable (IZK-NIZK)

Another remark which may be useful in applications, is that we can forward non-interactive proofs to propagate an (authentication) capability in the system, using (up to a constant number) of metaproof levels (if needed) or directly transfer the proof itself. Then, at a certain level, we reach the boundary and then the authentication may be needed but should not be transferred. Thus it is the right point to switch to an interactive proof. proving the possession of a NIZK based on the NIZK's and the statement. Recall that the possession of a metaproof is an elegant way to have a "zero-knowledge witness" of a statement.

5.5 Non-interactive witness-hiding equivalent to zero-knowledge.

The notion of Witness-hiding was originally suggested as a relaxation of the notion of zero-knowledge [FeSh]. In the non-interactive scenario, based on the existence of one-way function we can show, using the meta-proof system, that the notions are equivalent. Thus, our constructions can all be based on bounded non-interactive witness-hiding protocol (maybe such can be implemented based on any one-way function).

6 Conclusions

We have presented a few applications in various cryptographic scenarios based on recent Non-interactive Zero-Knowledge Proof-Systems we developed. The new tools can also be applied to produce Secure Distributed Computation tools and protocols, which we describe in [DeYu]. Details and full proofs will be provided in [DeYu] as well.

References

[BaMo] L. Babai and S. Moran, *Arthur-Merlin Games: A Randomized Proof System and a Hierarchy of Complexity Classes*, Journal of Computer and System Sciences, vol. 36, 1988, pp. 254–276.

[BeGo] M. Bellare and S. Goldwasser, *New Paradigms for Digital Signatures and Message Authentication based on Non-interactive Zero-knowledge Proofs*, Crypto 1989.

[BeMi] M. Bellare and S. Micali, *Non-interactive Oblivious Transfer and Applications*, Crypto 1989.

[BlDeMiPe] M. Blum, A. De Santis, S. Micali, and G. Persiano, *Non-Interactive Zero-Knowledge Proof Systems*, preprint.

[BlFeMi] M. Blum, P. Feldman, and S. Micali, *Non-Interactive Zero-Knowledge Proof Systems and Applications*, Proceedings of the 20th Annual ACM Symposium on Theory of Computing, Chicago, Illinois, 1988.

[Bl] M. Blum, *How to Prove a Theorem So No One Else Can Claim It*, Proceedings of the International Congress of Mathematicians, Berkeley, California, 1986, pp. 1444–1451.

[Co] S. A. Cook, *The Complexity of Theorem-Proving Procedures*, Proc. 3rd Ann. ACM Symp. on Theory of Computing, New York, pp. 151–158.

[De] Y. Desmeth, *Abuse-free Cryptosystems: Particularly Subliminal-Free Authentication and Signature*, preprint.

[DiHe] W. Diffie and M. E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, vol. IT-22, no. 6, Nov. 1976, pp. 644-654.

[DePe] A. De Santis and G. Persiano, *Public-Randomness in Public-key Cryptosystems*, Eurocrypt-90.

[DeMiPe1] A. De Santis, S. Micali, and G. Persiano, *Non-Interactive Zero-Knowledge Proof-Systems*, in "Advances in Cryptology - CRYPTO 87", vol. 293 of "Lecture Notes in Computer Science", Springer Verlag.

[DeMiPe2] A. De Santis, S. Micali, and G. Persiano, *Non-Interactive Zero-Knowledge Proof-Systems with Preprocessing*, Crypto 1988.

[DeYu] A. De Santis and M. Yung, *Non-Interactive Metaproofs and Non-Interactive Protocols*, Manuscript.

[EvGoMi] S. Even, O. Goldreich, and S. Micali, *On-line/Off-line Digital Signatures*, Crypto 1989.

[FeLaSh] U. Feige, D. Lapidot and A. Shamir, *Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String*, Focs 90.

[FeSh] U. Feige, and A. Shamir, *Witness-Hiding Protocols*, Proceedings of the 22th Annual ACM Symposium on Theory of Computing, 1990, announcement in Crypto-89.

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

[Go] O. Goldreich, *A Uniform-Complexity Treatment of Encryption and Zero-Knowledge*, Technical Report no. 568, Technion, June 1989.

[GoGoMi] O. Goldreich, S. Goldwasser, and S. Micali, *How to Construct Random Functions*, Journal of the Association for Computing Machinery, vol. 33, no. 4, 1986, pp. 792–807.

[GoMi1] S. Goldwasser and S. Micali, *Probabilistic Encryption*, Journal of Computer and System Science, vol. 28, n. 2, 1984, pp. 270–299.

[GoMiRa] S. Goldwasser, S. Micali, and C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, SIAM Journal on Computing, vol. 18, n. 1, February 1989.

[GoMiRi] S. Goldwasser, S. Micali, and R. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attack*, SIAM Journal of Computing, vol. 17, n. 2, April 1988, pp. 281–308.

[GoMiWi1] O. Goldreich, S. Micali, and A. Wigderson, *Proofs that Yield Nothing but their Validity and a Methodology of Cryptographic Design*, Proceedings of 27th Annual Symposium on Foundations of Computer Science, 1986, pp. 174–187.

[GoMiWi2] O. Goldreich, S. Micali, and A. Wigderson, *How to Play Any Mental Game*, Proceedings of the 19th Annual ACM Symposium on Theory of Computing, New York, pp. 218–229.

[Ha] J. Håstad, *Pseudorandom Generation under Uniform Assumptions*, Proceedings of the 22th Annual ACM Symposium on Theory of Computing, 1990.

[ImLeLu] R. Impagliazzo, L. Levin, and M. Luby, *Pseudo-Random Generation from One-way Functions*, Proceedings of 21st STOC, May 1989.

[ImNa] R. Impagliazzo and M. Naor, *Efficient Cryptographic Schemes Provably Secure as Subset Sum*, Proceedings of 30th FOCS, 1989.

[ImYu] R. Impagliazzo and M. Yung, *Direct Minimum Knowledge Computations*, in "Advances in Cryptology - CRYPTO 87", vol. 293 of "Lecture Notes in Computer Science", Springer Verlag pp. 40–51.

[LaSh] D. Lapidot and A. Shamir, These Proceedings.

[Na] M. Naor, *Bit Commitment using Pseudo-randomness*, Crypto 1989.

[NaYu] M. Naor and M. Yung, *Public-key Cryptosystems Probably Secure Against Chosen Ciphertext Attacks*, Proceedings of the 22th Annual ACM Symposium on Theory of Computing, 1990.

[Ro] J. Rompel, *One-way functions are Necessary and Sufficient for Secure Signatures*, STOC 90.

[Ya] A. Yao, *Theory and Applications of Trapdoor Functions*, Proc. 23rd IEEE Symp. on Foundations of Computer Science, 1982, pp. 80–91.