Linear Zero-Knowledge - A Note on Efficient Zero-Knowledge Proofs and Arguments

Ronald Cramer (ETH Zurich *) and
Ivan Damgård (Aarhus University † & BRICS ‡)

Abstract

We present a 4-move zero-knowledge proof system [21] for any NP language L, which allows showing that $x \in L$ with error probability less than 2^{-k} using communication corresponding to $O(|x|^c)+O(k)$ *bit commitments*, where c is a constant depending only on L. We also present a 4-move perfect zero knowledge interactive argument for any NP-language L. On input $x \in L$, the communication complexity is $O(|x|^c) \cdot max(k, l)$ *bits*, where l is the security parameter for the prover [1].

The protocols can be based on any bit commitment scheme with a particular set of properties. We suggest efficient implementations based on discrete logarithms or factoring.

As a function of the security parameters, our protocols have the smallest known asymptotic communication complexity among general proofs or arguments for NP. Moreover, the constants involved are small enough for the protocols to be practical in a realistic situation: our protocols allows proving/arguing satisfiability of a Boolean formula Φ (containing and-, or- and not-operators) with communication complexity at most $6n + 2$ commitments for the interactive proof and at most $5nl + 10l$ bits for the argument (assuming $k \leq l$), where n is the number of times Φ reads an input variable. Thus, if we use $k = n$, the number of commitments required for the proof is linear in n.

Finally, we present an application of our results that results in a protocol for oblivious transfer requiring $O(1)$ commitments of size $O(k)$ bits for a maximal cheating probability of 2^{-k}. Corresponding results for multiparty computations follow from this.

1 Introduction

Most known zero-knowledge interactive proofs or arguments for a general NP language, such as the classical methods of Goldreich, Micali and Wigderson [18] and Brassard, Chaum and Crépeau [5], yielding zero knowledge interactive proofs and -arguments for general NP-languages respectively, first construct a protocol that allows a prover to cheat with probability $1/2$, which is then iterated k times to achieve the required confidence level of $1/2^k$. These methods would require $\Omega(nk)$ bit commitments to show for instance satisfiability of a Boolean circuit of size n.

Several methods have been suggested for achieving the security amplification more efficiently than by the naive method. The work of Boyar, Brassard and Peralta [4] provides the first approach that improves on these communication complexities. They present zero knowledge proofs for circuit satisfiability using a "sub-quadratic number" of commitments. Roughly speaking, for large n and k they achieve zero knowledge interactive proofs using $O(\sqrt{n}k)$ bit commitments. Usually, one sets k equal to the size of the input, yielding roughly $O(n^{3/2})$ bit commitments in this case, hence a sub-quadratic number. Note that from this point of view, the results of [4] use a quadratic number of bit commitments.

Kilian [24] later extended their results by using the probabilistically checkable proofs (PCP) of [1]. More precisely, a zero-knowledge interactive proof that a circuit of size n is satisfiable is constructed using[2] $O(n^{1+c_1})+O(\log^{c_2}(n))k$ ideal bit commitments and having error probability 2^{-k}. For interactive arguments

*Institute for Theoretical Comp. Sc., ETH-Z, CH-8092 Zurich, Switzerland. Email: cramer@inf.ethz.ch. Research done while employed at CWI, Amsterdam, The Netherlands, and while visiting BRICS.

†Maths. & Comp. Sc. Dept.,Ny Munkegade, Aarhus, Denmark. Email: ivan@daimi.aau.dk

‡Basic Research in Computer Science, Center of the Danish National Research Foundation

[1]The meaning of l is that if the prover is unable to solve an instance of a hard problem of size l before the protocol is finished, he can cheat with probability at most 2^{-k}

[2]Here c_1 is any positive constant and $c_2 = O(1/c_1)$.

similar results are given, using a collision intractable hash function in addition. The latter result was further improved in [25], resulting in an interactive argument with communication complexity $O(lk \log l)$ bits. Here, and in the following, l is the security parameter for the prover. Thus, in order to cheat with probability larger than 2^{-k}, the prover must solve an instance of size l of a hard computational problem, such as finding a discrete logarithm modulo an l-bit prime[3].

In this work, we show that these communication complexities can be further improved. We do not use PCP's to build our protocols, in stead we use a new proof technique that may be of independent interest: We start from any Boolean formula Φ for checking an NP-witness for the language in question, and reduce the problem of showing that Φ is satisfiable to showing that a monotone formula constructed from Φ is satisfied by inputs contained in a given set of commitments. We then apply a technique derived from the "proofs of partial knowledge" introduced by Cramer et al. [8] (and independently in [27]). The properties we require from our bit-commitments are as follows: Of course, any bit commitment scheme must commit the prover, unconditionally or not, to a particular bit, and it must be impossible for the verifier, unconditionally or not, to find the bit committed to from a commitment. In addition, we need the following:

- Given a commitment C containing a bit b, the prover must be able to convince the verifier that C contains b, using an *honest verifier zero-knowledge* protocol. This protocol must be a *3-move Arthur-Merlin game*, it must have *exponentially small* error probability, and have communication complexity corresponding to a *constant* number of commitments.

As detailed later, this conditions can be met by commitments based on the discrete logarithm problem in a group of prime order, or the factoring problem.

Given a family $\mathcal{C}$ of polynomially sized Boolean circuits, one can efficiently transform each $C \in \mathcal{C}$ to a Boolean formula Φ_C such that Φ_C is satisfiable if and only if C is. Moreover, $|\Phi_C| = O(|C|)$. Using this, our approach results in zero-knowledge interactive proofs for circuit satisfiability (and thus for NP) with error probability 2^{-k} and communication complexity corresponding to $O(n)+O(k)$ commitments; for interactive arguments for NP we get communication complexity $O(n) \cdot max(k,l)$ bits (we count commitments for the proof and bits for the argument to facilitate comparison with [24, 25]).

Comparing this to [24], [25] which were the best results so far, we see that for interactive proofs, the term depending on k has been reduced from $O(\log^{c_2} n)k$ to k. For arguments, our result is inferior to [25] when viewed as a function of n, but superior as a function of the security parameters k and l. Note that our interactive argument has no need for a collision-intractable hash function, we only need commitments with the right properties. Hence our cryptographic assumption is potentially weaker than the ones needed in [25].[4]

In a different line of research, Kilian et al. propose in [22] a method for using a multibit commitment scheme in a protocol such as the one for circuit SAT in [5]. If this multibit commitment is such that the amortized communication needed per bit committed to is small, this can lead to a more communication efficient protocol than [5] with a one-bit commitment scheme. This method is essentially a way to execute more efficiently the atomic step of a protocol such as the one in [5]. The error reduction is still obtained by simple sequential iteration. Therefore improvements obtained by this method will in general be inferior to ours as a function of the parameter k controlling the error reduction.

Very recently, Damgård and Pfitzmann [11] have shown that the multi-bit commitment scheme of Damgård et al. [12] based on collision intractable hashing can be used with the method of [22]. This leads to an interactive argument which has in a practical situation communication complexity similar to ours. However it is only statistical zero-knowledge and needs a linear, rather than a constant number of rounds. Another possibility, leading to an interactive proof, is to use the commitment scheme of Naor [23] based on pseudorandom generators.

If one adopts the usual convention of setting the security parameter k equal to the input size, our result implies a zero knowledge interactive proof that proves satisfiability of a circuit of size n with error probability 2^{-n} using $O(n)$ *commitments*. Even if an extremely small PCP would exist, the protocol in [24] would use $\Omega(n \log^{c_2} n)$ *commitments* to solve the same problem. To the best of our knowledge, our protocol is the first to achieve "linear zero-knowledge" in this sense. For arguments, we get $O(n^2)$ *bits* using $l = k = n$, where [25] would be $O(n^2 \log n)$ *bits*.

Our final result applies our general result and our concrete bit commitment example to build a protocol for oblivious transfer requiring $O(1)$ commitments of size $O(k)$ bits for a maximal cheating probability of 2^{-k}. Corresponding results for multiparty computations follow from this.

[3] More precisely, one can show that if a prover can argue a false statement with success probability $\epsilon > 2^{-k}$, then he can solve the hard problem in time polynomial in $1/(\epsilon - 2^{-k})$.

[4] Although no example is currently known that would support our needs, and not simultaneously imply a collision intractable hash function.

Remark

For the case of interactive proofs, we have, like [24], ignored in the statement of results the communication needed to set up the commitment scheme [5]. This is reasonable, as the same commitment scheme can be reused in many proofs. For arguments, however, an attractive point is that cheating is only possible if the intractability assumption used is broken *while the protocol is running* [6]. This, however, is only true if a new instance of the commitment scheme is chosen in every run of the protocol. Our communication complexity for arguments therefore includes communication for setting up the commitment scheme.

2 Technical Ingredients

Our main result uses *partial proofs* and *bitcommitments* with special properties. These are explained hereafter.

2.1 Partial Proofs

We will now state a result implied by [8]. For convenience, we re-formulate it to match our context here.

Σ-protocols

Let a Σ-protocol (A, B) be a three move interactive protocol between a probabilistic polynomial time prover A and a probabilistic polynomial time verifier B, where the prover acts first. The verifier is only required to send random bits as a challenge to the prover.

More precisely, let $R = \{(\alpha, \beta)\}$ be a binary relation and assume that for some given polynomial $p(\cdot)$ it holds that $|\beta| \leq p(|\alpha|)$ for all $(\alpha, \beta) \in R$. Furthermore, let R be testable in polynomial time, and let R^* denote the collection of strings α such that, for some string β, $(\alpha, \beta) \in R$. The string β is called a witness for α. For some $(\alpha, \beta) \in R$, the common input to both players is α while β is private input to the prover. For such given α, let (a, c, z) denote the conversation between the prover and the verifier. To compute the first and final messages, the prover invokes efficient algorithms $a(\cdot)$ and $z(\cdot)$, respectively, using (α, β) and random bits as input. Using an efficient predicate $\phi(\cdot)$, the verifier decides whether the conversation is accepting with respect to α. The relation R, the algorithms $a(\cdot)$, $z(\cdot)$ and $\phi(\cdot)$ are public. The length of the challenges is denoted t_B, and we assume that t_B only depends on the length of the common string α.

[5] In any real implementation, the verifier needs to receive some public parameters of the commitment scheme, and possibly a zero-knowledge proof that they were chosen correctly

[6] in contrast to the situation for proofs, where breaking the assumption at any later time can cause problems

In the present context, we will assume that we are given Σ-protocols satisfying the following security properties. First, (A, B) satisfies a strong flavour of knowledge soundness: Let (a, c, z) and (a, c', z') be two conversations, that are accepting for some given α. If $c \neq c'$, then $\alpha \in R^*$ and, on input α and those two conversations, we can efficiently compute β such that $(\alpha, \beta) \in R$. This is called *special soundness*, and the pair of accepting conversations (a, c, z) and (a, c', z') with $c \neq c'$ is called a *collision* . Finally, we assume (A, B) to satisfy *special honest verifier zero knowledge*. This means that we are given a (probabilistic polynomial time) simulator M that on input $\alpha \in R^*$ generates accepting conversations with the same distribution as when A and B execute the protocol on common input α (and A is given a witness β for α), and B indeed honestly chooses its challenges uniformly at random. The simulator is special in the sense that it can additionally take a random string c as input, and output an accepting conversation for α where c is the challenge.

Monotone Function Families

A function $f : \{0,1\}^n \rightarrow \{0,1\}$, $f \not\equiv 0, 1$, is called *monotone* if the following holds. If $f(x_1 \ldots x_n) = 1$ and if $y_1 \ldots y_n \in \{0,1\}^n$ is such that, for $i = 1 \ldots n$, $y_i = 1$ if $x_i = 1$, then $f(y_1 \ldots y_n) = 1$. By $\mathcal{F}_n$ we denote a family of monotone functions where each of its members takes n bits as input. $\mathcal{F} = \cup_{n>0}\mathcal{F}_n$, denotes the union of such collections: a family of monotone functions. In the notation $f_n \in \mathcal{F}$, the subscript n to f_n serves as a reminder that $f_n \in \mathcal{F}_n$. Although this is not reflected in our terminology, we will only consider families $\mathcal{F}$ of monotone functions where its of functions f_n can be computed in time polynomial in n. Furthermore, we will assume that membership of $\mathcal{F}$ can be efficiently decided.

Let $J \subset \{1, \ldots, n\}$. We define $x_J \in \{0,1\}^n$ by setting the i-th position in x_J to 1 if $i \in J$ and to 0 otherwise. Then $f_n(J)$ denotes $f_n(x_J)$.

By a *monotone Boolean formula* we mean a function given as a Boolean formula consisting of AND-operators and OR-operators only. A family $\mathcal{F}$ of monotone Boolean formulas is polynomially sized if the number of operators is polynomially bounded in n.

Partial Proofs Sufficient for Our Context

Let $\mathcal{F}$ be a family of monotone functions and let R be a binary relation as before. For all n, for all $f_n \in \mathcal{F}_n$ and for all l, consider the collection of all tuples $\alpha = (f_n, \alpha_1, \ldots, \alpha_n)$ such that $\alpha_i \in \{0,1\}^l$ for $i = 1 \ldots n$. Let $\beta = \{\beta_j\}_{j \in I} \subset \{0,1\}^*$ be given where $I \subset \{1, \ldots, n\}$ and $|\beta_j| \leq p(|\alpha_j|)$ for $j \in I$. Then $(\alpha, \beta) \in R_{\mathcal{F}}$ if and only if $f_n(I) = 1$ and $(\alpha_j, \beta_j) \in R$

for $j \in I$. Note that by our assumptions on R and $\mathcal{F}$, the composite binary relation can be tested efficiently. The results from [8] imply the following theorem (see also [9] for the full version).

THEOREM 1 *Let (A, B) be a Σ-protocol for relation R, satisfying special honest verifier zero-knowledge and special soundness. Let $\mathcal{F}$ be a polynomially sized family of monotone Boolean formulas Then there exists a Σ-protocol (A', B') for relation $R_{\mathcal{F}}$ satisfying special honest verifier zero-knowledge and special soundness. The size $t_{B'}$ of the challenges in (A', B') is equal to t_B. If each formula $f_n \in \mathcal{F}$ reads each input bit only once, then the communication complexity of (A', B') is n times that of (A, B) plus $t_{B'}$ bits.*

By a standard rewinding argument, we have the following. For definitions of proofs of knowledge and knowledge extractors, see Bellare and Goldreich [3].

COROLLARY 1 *Suppose that the input words α to (A, B) have length l bits, and that the challenge length t_B is equal to l as well. Then (A', B') is a proof of knowledge. Let A^* denote a prover that is successful with probability $\epsilon > 2^{-l}$ in time T_{A^*}. A knowledge extractor runs in expected time polynomial in T_{A^*} and $1/(\epsilon - 2^{-l})$.*

2.2 Commitments with Linear Proof of Contents

Bit commitment schemes of the kind we use consist of functions $\texttt{commit} : \{0,1\}^{l_r} \times \{0,1\} \rightarrow \{0,1\}^l$ and $\texttt{verify} : \{0,1\}^l \times \{0,1\}^{l_r} \times \{0,1\} \rightarrow \{accept,\ reject\}$, whose descriptions are output by a probabilistic polynomial time algorithm G on input 1^l, where l is a security parameter. The value l_r is polynomially bounded in l. For our purposes here, we require that $l_r = O(l)$. We refer to `commit` and `verify` as the *public key* of the commitment scheme. To commit to a bit b, one chooses r at random from $\{0,1\}^{l_r}$ and computes the commitment C as $C \leftarrow \texttt{commit}(r, b)$. The value r masks the bit b. To verify whether a commitment has been opened correctly, one verifies whether $\texttt{verify}(C, r, b) = accept$.

For interactive proofs, we will need bit commitments to be *unconditionally binding.* This means that the bit b is uniquely determined from the commitment C. Of course we also need the scheme to hide the bit, but the best we can get in this case, is that it is *computationally hiding*: the distributions of commitments to 0 and and to 1, respectively, are computationally indistinguishable: no probabilistic polynomial time algorithm receiving as input a commitment to 0 or 1, can guess the bit b with probability significantly better than 1/2.

For interactive arguments, we will use bit commitment schemes with dual properties: *unconditionally hiding.* This means that the distributions of commitments to 0 and and to 1, respectively, are identical. Now, with respect to the binding property, the best we can achieve is that the scheme is computationally binding. This means that no probabilistic polynomial time algorithm can compute a commitment that can be opened in both ways: it is infeasible to compute $C \in \{0,1\}^l$, and $r_0, r_1 \in \{0,1\}^{l_r}$ such that $\texttt{verify}(C, r_0, 0) = \texttt{verify}(C, r_1, 1) = accept$, except with negligible probability.

Unconditionally hiding commitments may in addition be *trapdoor* [5] (also called *chameleon*). For a trapdoor commitment, the generator G outputs in addition a string T called the *trapdoor information.* Given the trapdoor, one can cheat the commitment scheme. More formally, there is a polynomial time algorithm that on input T will produce pairs r_0, r_1 such that $\texttt{commit}(r_0, 0) = \texttt{commit}(r_1, 1) = C$, $\texttt{verify}(C, r_0, 0) = \texttt{verify}(C, r_1, 1) = accept$, and the distribution of C is the same as that of $\texttt{commit}(r, b)$ for random r. We will assume that, on the other hand, given C and any pair r_0, r_1 such that $\texttt{verify}(C, r_0, 0) = \texttt{verify}(C, r_1, 1) = accept$, it is easy to compute T. Note that by the binding property, it is infeasible to compute the trapdoor information T given just the public key of the commitment scheme. Finally, for an unconditionally hiding trapdoor commitment scheme, we require that there exists a linear (i.e., with communication complexity linear in l), witness hiding [15] proof of knowledge of the trapdoor T.

When $C = \texttt{commit}(r, b)$ is a commitment, we define $\neg C$ to be special symbol $\neg$ followed by the string $\texttt{commit}(r, b)$. And we extend the verify function such that $\texttt{verify}(\neg C, r, b) = \texttt{verify}(C, r, 1 - b)$. In other words, $\neg C$ is a commitment that is opened by opening the *basic* commitment C and negating the resulting bit. In the following, unless otherwise stated, commitments may be either basic or negated as described here.

A bit commitment scheme has a *linear proof of contents*, if there is a Σ-protocol (A, B) taking as input (basic) commitment $C = \texttt{commit}(r, b)$ and bit b and with the following properties:

1. (A, B) is a *proof of knowledge*, satisfying *special soundness*, that A knows how to open C as a commitment to b. More precisely, from two conversations that constitute a collision, one can efficiently compute r such that $C = \texttt{commit}(r, b)$.
2. (A, B) is *special honest verifier perfect zero knowledge*, with simulator M.
3. The *size* of the conversation is $O(l)$ bits and the challenge size t_B is linear in l. By some standard manipulation techniques, we may assume that $t_B = l$, while the size of the conversation is still $O(l)$ bits.

Note that since commitments to b are assumed to be indistinguishable from commitments to $1-b$, the simulator M should output an accepting conversation on input $C = \texttt{commit}(r,b), 1-b$, except with negligible probability. We will assume for simplicity that it always does so (this holds in our concrete examples).

Regarding the *existence* of the required bit commitments, we note the following. In the Section 4, we give two example bitcommitment schemes with the properties stated above: an unconditionally binding one and another that is unconditionally hiding. Both are based on the difficulty of computing discrete logarithms. It turns out (see [9]) that we can construct the commitment schemes required for arguments, under the assumption that a family of *special one-way group isomorphisms* exists. Let f be a one-way isomorphism between groups K and L. Such a function is called special if we can efficiently compute a (large) prime T such that for all $y \in L$ we can efficiently compute x with $f(x) = y^T$. Such functions f can be realised under both the discrete logarithm and RSA assumptions. Also, it is possible to realize a suitable unconditionally binding scheme based on the factoring problem, more precisely speaking, based on the r-th residuosity problem.

3 Main Result

3.1 General Approach

We start by presenting a general method for constructing a *communication efficient* perfect *honest verifier* zero knowledge proof (A', B') that a given word x is a member of an NP-language L. Given (a family of) Boolean formulas Φ that verify witnesses for L and a bit commitment scheme with negation and linear proof of contents (A, B), we construct (a family of) monotone formulas Φ' and invoke Theorem 1. The prover P will commit to a witness w for x, after which the prover P and verifier V will run the protocol (A', B'), P running A', and V running B' as subroutines, respectively. P will only be accepted by V if the bits committed to constitute a witness for x, i.e, $x \in L$. This results in interactive proofs and arguments for L that are *honest verifier* zero knowledge.

Then in the following two sections, we show how to obtain zero-knowledge in general for these interactive proofs, resp. arguments. For interactive proofs, we will require them bitcommitments to be unconditionally binding, while for arguments we require them to be unconditionally hiding.

Let an input word $x \in L$ of length k bits be given, and let Φ be a Boolean formula verifying a witness for x. Without loss of generality we may assume that Φ consists of AND-, OR- and NOT-operators only, with the NOT-operators occurring at the inputs. Let m denote the number of different input variables to Φ, and let n denote the number of times that Φ reads an input variable. A monotone formula Φ' is obtained from Φ by removing the negations and by renaming the input variables such that all n references to the inputs refer to different variables. For example, if $\Phi = (a \wedge b) \vee (\neg a \wedge \neg b)$, then we would have $\Phi' = (a \wedge b) \vee (c \wedge d)$.

Let Ψ be any monotone formula on n input variables, and let a set of commitments $D_1, .., D_n$ be given. Then we say that the set of strings $r_1, ..., r_n$ Ψ-*opens* $D_1, ..., D_n$ if $\Psi(\gamma_1, ..., \gamma_n) = 1$, where $\gamma_i = 1$ if and only if $\texttt{verify}(D_i, r_i, 1) = accept$. Let R be the binary relation consisting of all pairs (D, r) such that $\texttt{verify}(D, r, 1) = accept$. Note that the linear proof of contents (A, B) that comes with our bit commitment schemes immediately gives a proof of knowledge for the relation R: to show that basic commitment C contains 1, run (A, B) on input $C, 1$, to show that $\neg C$ contains 1, run (A, B) on input $C, 0$.

Recall that we have assumed that both the challenge size t_B is equal to l and the size of conversations in (A, B) are linear in l. Here, l is the size of a commitment. Taking into account that (A, B) also satisfies special soundness and special honest verifier zero knowledge, we have by Theorem 1 and Corollary 1:

PROPOSITION 1 *Let $\mathcal{F}$ be a family of monotone Boolean formulas of polynomial size, such that for each n, each $f_n \in \mathcal{F}$ reads each of its n input bits exactly once. Let a bit commitment scheme be given which has a linear proof of contents (A, B). If commitments $D_1, ..., D_n$ of size l and $\Psi \in \mathcal{F}_n$ are given, then there exists an honest verifier zero knowledge Σ-protocol (A', B') showing that A' can Ψ-open $D_1, ..., D_n$. Furthermore, from two conversations of (A', B') of form $(a, c, z), (a, c', z')$, where $c \neq c'$ one can efficiently compute a set of strings that Ψ-opens $D_1, ..., D_n$. Thus (A', B') is a proof of knowledge for relation $R_{\mathcal{F}}$, satisfying special soundness. The communication complexity is l bits plus n times that of (A, B). This corresponds to $O(|\Psi|)$ bit commitments of size l.*

COROLLARY 2 *Suppose that A^* is a prover accepted by the honest verifier B' with probability $\epsilon > 2^{-l}$. Then there exists a probabilistic algorithm Ext that outputs set of strings that Ψ-opens $D_1, ..., D_n$, running A^* as a rewindable blackbox, with expected running time polynomial in T_{A^*} and $1/(\epsilon - 2^{-l})$, where T_{A^*} denotes A^*'s running time.*

We now consider the following honest verifier zero knowledge protocol (P', V') for showing that Φ is satisfiable.

Step 1 : Let $x \in L$ and let a witness $w = (w_1, .., w_m)$ be given by input bits that satisfy Φ. For $i = 1 \ldots m$, P' now computes basic commitments C_i for these bits w_i: P' puts $C_i \leftarrow \texttt{commit}(r_i, w_i)$, where r_i is chosen at random from $\{0,1\}^{l_r}$. P' sends these C_i's to V'.

Step 2 : Number the positions in Φ where an input bit is used (at the input wires) from 1 to n. For $j = 1 \ldots n$, let $D_j = C_i$, if the bit w_i is used at this position, and let $D_j = \neg C_i$ if the bit $1 - w_i$ is used.

Step 3 : Using the protocol (A', B') guaranteed by Proposition 1, P' now convinces V that that the bits contained in $D_1, ..., D_n$ satisfy the monotone formula Φ' (that is, $D_1, ..., D_n$ can be Φ'-opened). Here P' plays the role of A' and V' plays the role of B'.

The protocol (P', V') gives rise to the following theorems.

THEOREM 2 *Suppose there exists an unconditionally binding bit commitment scheme which has a linear proof of contents. Then any $L \in NP$ has a constant-round honest verifier computational zero-knowledge interactive proof system that proves $x \in L$ with error probability at most 2^{-k} using a total of $O(|x|^c)$ commitments, for some constant c depending only on L.*

THEOREM 3 *Suppose there exists an unconditionally hiding trapdoor bit commitment scheme which has a linear proof of contents. Then any $L \in NP$ has a constant-round honest verifier perfect zero-knowledge interactive argument that $x \in L$, with communication complexity $O(|x|^c) \cdot max(k, l)$ bits, where c is a constant depending only on L. If a prover P^* can cheat with probability $\epsilon > 2^{-k}$ in time T_{P^*}, the prover can break instances of the commitment scheme of size l with expected running time time polynomial in T_{P^*} and $1/(\epsilon - 2^{-k})$.*

PROOF. In case of interactive proofs, we set $k = l$, and execute (P', V'). For interactive arguments, we execute (P', V') s times in parallel, where s is minimal such that $sl \geq k$. The security properties are invariant under parallel composition. *Completeness* is trivial. As for *soundness*, note that if a prover P^* has probability of success greater than 2^{-k}, then there exist $\rho_1, \ldots, \rho_n$ that Φ'-open $D_1, \ldots, D_n$. Thus, $\Phi'(\gamma_1, \ldots, \gamma_n) = 1$ where $\gamma_j = 1$ if and only if $\texttt{verify}(D_j, \rho_j, 1) = accept$. Recall that by definition, each D_j is equal to C_i or $\neg C_i$ for some i. Let V_i denote the set of indices j, $1 \leq j \leq n$, such that D_j was set equal to C_i, and let V_i' similarly denote those where D_j was set equal to $\neg C_i$. Define the set W as the set of indices j, $1 \leq j \leq n$, with $\gamma_j = 1$. We define the bits w_i and w_i', $i = 1 \ldots m$, by setting $w_i = 1$ if $V_i \cap W \neq \emptyset$ and 0 otherwise, and $w_i' = 1$ if $V_i' \cap W \neq \emptyset$ and 0 otherwise. Note that $w_i = w_i' = 1$ implies that we can find j and j' such that $\texttt{verify}(C_i, \rho_j, 1) = \texttt{verify}(\neg C_i, \rho_{j'}, 1) = accept$. If we assume that at least one of w_i and w_i' is equal to 0 for each i, and put $w_i = 1$, $w_i' = 0$ instead in those cases where $w_i = w_i' = 0$, it is easy to see that $w = (w_1, \ldots, w_m)$ satisfies Φ. In case of interactive proofs, where an unconditionally binding scheme is used, the case $w_i = w_i' = 1$ is impossible, so the error probability is at most 2^{-k}. Now for the case of interactive arguments where unconditionally hiding bit commitments are used, the case $w_i = w_i' = 1$ implies that the prover is breaking the binding property of the commitment scheme: by definition $\texttt{verify}(C_i, \rho_j, 1) = \texttt{verify}(\neg C_i, \rho_{j'}, 1) = accept$ implies that C_i can be opened as 0 and 1. Since we may view (P', V') as a Σ-protocol satisfying special soundness with challenge size $sl \geq k$, we have similarly to Corollary 2 that the prover can break the commitment scheme with expected time running time polynomial in T_{P^*} and $1/(\epsilon - 2^{-k})$, if the success probability ϵ is greater than 2^{-k}. The latter argument also shows that (P', V') is a proof of knowledge in both cases of interactive proofs and arguments. Concerning *honest verifier zero knowledge* simulation, we construct the C_i's as a set of all-0 commitments, and compute the D_j's from this. We then invoke (s times, in case of the arguments) the honest verifier simulator of (A', B'). This simulation is perfect for unconditionally hiding commitments, and is computationally indistinguishable for unconditionally binding commitments. Finally, the *communication complexities* are argued as follows. Note that we may assume that $|\Phi| = O(|x|^c)$, where the constant c only depends on the language L. We also have $n = O(|x|^c)$, and that the communication complexity of (P', V') is that of (A, B), except that for arguments it is repeated s times. Therefore, in the case of interactive proofs, we need communication corresponding to $O(|x|^c)$ bitcommitments of size k (since we have put $k = l$), and for arguments, we need some $\lceil k + 1/l \rceil \cdot O(|x|^c)$ bitcommitments of size l. Since we are counting *bits* in this case, the communication is $O(|x|^c) \cdot max(k, l)$ bits.

3.2 Zero Knowledge Interactive Proofs for NP

The problem that (P', V') is only honest verifier zero-knowledge can be solved in two ways: One can use the general transformation from [10], the basic idea of which goes back to [2]. Here the prover and verifier do two-party coinflipping to determine the challenge to be answered by the prover. This requires only the unconditionally binding commitments that we already assumed. If in addition an unconditionally hiding commitment scheme is available, one can in stead use a method due to Goldreich and Kahan [20], namely to let the verifier commit to the challenge before (P', V') is executed. This turns (P', V') into a constant-round zero knowledge proof system for L. Some commitment schemes (including our examples) allow the verifier to commit to the entire k-bit challenge in one commitment of size $O(k)$ bits [7] This leads to the following results:

THEOREM 4 *Suppose there exists an unconditionally binding bit commitment scheme which has a linear proof of contents. Then any $L \in NP$ has a computational zero-knowledge interactive proof system that proves $x \in L$ with error probability at most 2^{-k} using a total of $O(|x|^c) + O(k)$ commitments of size $O(k)$ bits, for some constant c depending only on L.*

THEOREM 5 *Assumption as in in Theorem 4, but assume in addition that there exists an unconditionally hiding bit commitment scheme allowing commitment to k bits in a commitment of size $O(k)$ bits. Then any $L \in NP$ has a 4-move computational zero-knowledge interactive proof system that proves $x \in L$ with error probability at most 2^{-k} using a total of $O(|x|^c)$ commitments of size $O(k)$ bits, for some constant c depending only on L.*

3.3 Interactive Arguments for NP

To build a zero-knowledge interactive argument (P, V) from (P', V'), we use an unconditionally hiding *trapdoor* bit commitment scheme with an *efficient witness hiding proof of knowledge* of the trapdoor.

Step 1 : The verifier V runs the key generator G, sends the resulting public key for the commitment scheme to the prover P, and keeps the trapdoor T private.

Step 2 : The verifier V gives a witness hiding proof of knowledge of the trapdoor to the prover P.

Step 3 : Protocol (P', V') is executed using the commitment scheme instance just generated, where P and V play the roles of P' and V', respectively.

The idea is taken from [16]. The protocol as shown here has 6 moves, but this can be condensed to 4 moves in the same way as in [16]. The proof of soundness remains essentially the same, but note that in order to fool V, the prover P still has to break the commitment scheme, as follows from the proof of Theorem 3. In the case of *trapdoor* commitments we have required that breaking the commitment scheme is essentially as difficult as finding the trapdoor T. However, the verifier's witness hiding proof does not help to do that. Hence the soundness of (P', V') is preserved. Furthermore, the protocol is now zero-knowledge, since the simulator can use the knowledge extractor for the verifier's proof of knowledge to get the trapdoor information [8]. Given the trapdoor, simulation of the rest of the protocol is trivial. The witness hiding proof costs, by assumption on the commitment scheme, a communication complexity that is $O(l)$ bits. We then get the following by inspection of (P, V):

THEOREM 6 *Suppose there exists an unconditionally hiding trapdoor bit commitment scheme which has a linear proof of contents and a linear, witness hiding proof of knowledge of the trapdoor. Then any $L \in NP$ has a 4-move perfect zero-knowledge interactive argument that $x \in L$, with communication complexity $O(|x|^c) \cdot max(k, l)$ bits, where c is a constant depending only on L. If a prover P^* can cheat with probability $\epsilon > 2^{-k}$ in time T_{P^*}, the prover can break instances of the commitment scheme of size l with expected running time polynomial in T_{P^*} and $1/(\epsilon - 2^{-k})$.*

4 Concrete Bitcommitment Schemes

We present two bit commitment schemes with properties as required in main results. Our examples have the additional property that commitments can be *negated*: the verifier can, on his own, compute from a commitent C to b a new basic commitment C' to $1 - b$. Because of this, for the linear proof of contents, it suffices with a protocol for showing that a commitment contains 1.

Scheme I, described below, is an *unconditionally binding* bit commitment scheme based on the discrete logarithm problem in a group of prime order. It is derived from the Diffie-Hellman/El Gamal encryption scheme [13, 14]. For concreteness, we think here of this group as a subgroup of $\mathbb{Z}_p^*$, where p is a prime and the prime q divides $p - 1$. But any group of order q would do, such as an elliptic curve group.

Key Generation : The key generator G for this scheme chooses large primes p and q such that $q | p - 1$, and a random element $g \in \mathbb{Z}_p^*$ of order q. Next h and w are chosen at random in the group generated by g. The public key of the commitment scheme is (p, q, g, h, w).

Commitment : The function `commit` is defined as $C \equiv \texttt{commit}(r, b) \leftarrow (g^r \bmod p, w^b h^r \bmod p)$, where r is chosen at random from $\mathbb{Z}_q$.

Opening : To open the commitment C, the values r and b are revealed. The algorithm $\texttt{verify}(C, r, b)$

[7] This is done in our scheme from Section 4 by simply replacing the bit b committed to by any value modulo q.

[8] Although this by itself may not produce the trapdoor with absolute certainty, the simulator can run an exhaustive search for the trapdoor in parallel with the extractor. This will then produce the trapdoor in those exponentially few cases where the extractor fails, while the expected running time remains polynomial

outputs *accept* if and only if ($g^r \bmod p, w^b h^r \bmod p$).

Negation : Given a commitment $C = \texttt{commit}(r, b) \equiv (\gamma_1, \gamma_2)$, then $C' = (\gamma_1^{-1}, w\gamma_2^{-1})$ is a commitment to $1 - b$.

It follows immediately that this commitment scheme is *unconditionally binding.* Under a standard assumption about Diffie-Hellman/El Gamal encryption the scheme provides *computational hiding* of the bit committed to. Finally, the following protocol (A, B), which is an adaptation of Schnorr's protocol [28], is sufficient for a *linear proof of contents.*

Move 1 : Let $C = (g^r \bmod p, wh^r \bmod p)$ be a commitment to $b = 1$, and let C be common input to A and B. Let $C' \equiv (\delta_1, \delta_2)$ be the negation of C. The value r is private input to A. Then, A chooses f at random from $\mathbb{Z}_q$ and computes $a_1 \leftarrow g^f \bmod p$ and $a_2 \leftarrow h^f \bmod p$. A sends a_1 and a_2 to B.

Move 2 : B chooses c at random from $\mathbb{Z}_q$ and sends it to A.

Move 3 : A computes $z \leftarrow f - cr \bmod q$ and sends z to B. Finally, B checks that $g^z = a_1\delta_1^c \bmod p$ and $h^z = a_2\delta_2^c \bmod p$

Scheme II concerns a bit commitment scheme that is *unconditionally hiding* [9]. It is also based as on exponentiation in a group of prime order, where computing discrete logarithms is hard.

Key Generation : Choose two large primes p and q such that $q|p-1$. Then select at random two elements $g_1, g_2 \in \mathbb{Z}_p^*$ of order q. Finally, choose w_1, w_2 at random from $\mathbb{Z}_q$ and compute $h \leftarrow g_1^{w_1} g_2^{w_2} \bmod p$. The public key is (p, q, g_1, g_2, h).

Commitment : The algorithm $\texttt{commit}$ is defined by $C \equiv \texttt{commit}((r_1, r_2), b) \leftarrow g_1^{r_1} g_2^{r_2} h^b \bmod p$, where r_1, r_2 are chosen at random from $\mathbb{Z}_q$.

Opening : To open a commitment C, the values r_1, r_2 and b are revealed. The algorithm $\texttt{verify}(C, (r_1, r_2), b)$ outputs *accept* if and only if $C = g_1^{r_1} g_2^{r_2} h^b \bmod p$.

Negation : Given a commitment $C = \texttt{commit}((r_1, r_2), b)$, then $C' \equiv C^{-1}h \bmod p = \texttt{commit}((-r_1, -r_2), 1-b)$ is a commitment to $1-b$.

Trapdoor : A trapdoor can be any pair u_1, u_2 such that $h = g_1^{u_1} g_2^{u_2} \bmod p$. A *witness hiding proof of knowledge of the trapdoor*, with linear communication complexity, is given by Okamoto's scheme [26].

[9] The unconditionally hiding multi-bit commitment, derived from Scheme II as indicated in Section 3.2, is referred to as *Scheme II'* in the following.

The scheme has the following *linear proof of contents* (A, B), taken from [26].

Move 1 : Let $C = g_1^{r_1} g_2^{r_2} h \bmod p$ be a commitment to $b = 1$, and let C be common input to A and B. The values r_1 and r_2 are private input to B. Let δ denote the negation $C^{-1}h \bmod p$ of C. Now A chooses f_1, f_2 at random from $\mathbb{Z}_q$, computes $a \leftarrow g_1^{f_1} g_2^{f_2} \bmod p$ and sends a to B.

Move 2 : B chooses c at random from $\mathbb{Z}_q$ and sends it to A.

Move 3 : A computes $z_1 \leftarrow f_1 - cr_1 \bmod q$ and $z_2 \leftarrow f_2 - cr_2 \bmod q$, and sends z_1 and z_2 to B. Finally, B checks that $g_1^{z_1} g_2^{z_2} = a\delta^c \bmod p$.

5 Concrete Communication Complexities

Assume we use one of the two example commitment schemes shown in Section 4 to implement our protocols. To evaluate their practical potential, we compute the exact communication complexities that result. For both commitments schemes, we have to choose the parameter l such that computing discrete logarithms is infeasible, which means that l should be $700 - 1000$. Note that having l of that length and having just one iteration of (P', V'), yields an error probability of at most $1/2^{700}$, so we may assume that one iteration suffices and that $k \leq l$ in most practical situations. In general we can say that in the worst case, the formula Φ uses every input bit and its complement exactly once, so that no reuse of commitments is possible. With these assumptions and counting for convenience one number modulo p as one commitment, we get by inspection of the protocols:

PROPOSITION 2 *Suppose the zero-knowledge interactive proof from Theorem 5, resp. the perfect zero-knowledge argument from Theorem 6, is executed using commitment schemes I and II' from Section 4, resp. commitment scheme II, then assuming that $l \geq k$, the communication complexities will be at most $6n + 2$ commitments (of size l bits), resp. $5nl + 10l$ bits, where n is the number of times a Boolean formula for verifying an NP witness for L reads an input variable.*

A simple computation shows that with for example about 6 Mbyte of communication, and using $k = 50, l = 768$, n can be up to about 10.000. This might be enough to prove, for instance, that you know a DES key encrypting a given cleartext block to a given ciphertext block.

We note that in general, our protocol may be significantly optimized by building ad hoc as small a formula Φ as possible for the problem. Furthermore it may be possible to find a smaller monotone formula computing the same function as the one constructed directly from Φ.

6 An Application: Oblivious Transfer and Multiparty Computations

Loosely speaking, the multiparty computation problem is defined by a function f with p arguments and p participants, such that the i'th participant owns a value x_i of the i'th argument to f. The goal is to design a protocol such that all participants learn the value $f(x_1, ..., x_p)$, but no coalition of participants can, even by deviating from the protocol, learn more about the inputs that what is implied by the *own* inputs and the result.

The classical protocols for solving this problem in the model where only broadcast is available for communication can be found in [19], [29], with efficiency improvements e.g. in [6]. Here we present a much more substantial improvement by using our commitment scheme from Section 4 to implement a fundamental primitive known as *Committed Oblivious Transfer* (COT). An efficient COT protocol automatically leads to efficient multiparty computation protocols by known reductions. A concrete one, the basic idea of which goes back to [19], can be found in [7].

A comitted oblivious transfer takes place between two parties S and R. Initally S has made two commitments A_0, A_1 containing bits a_0, a_1, and R has made commitment B to bit b. The purpose of the protocol is that R should end up making a commitment C to bit a_b. This must be done under the conditions that A does not learn b, and B is forced to commit to a_b, but does not learn a_{1-b}. We have the following result:

THEOREM 7 *If the commitment scheme from Section 4 is computationally hiding, then there is a protocol that implements committed oblivious transfer between polynomially bounded parties, with error probability 2^{-k} and communication complexity corresponding to $O(1)$ commitments of size $O(k)$ bits.*

A formal treatment of our COT protocol and its application to multiparty computations, in particular the distinction between static and passive adversaries would far exceed the space limitation of this paper. Here, we only give the protocol and an informal sketch of a proof for it. Note that the commitments from section 4 form a multiplicative group by componentwise multiplication modulo p. In the following, when C is a commitment, let $\tilde{C}$ denote the bit contained in C. The idea of the protocol is that R can "blind" a commitment from S by multiplying it by something random, hence S can open the result without knowing which of two commitment he is actually opening. Our general interactive proofs for satisfiability of a Boolean formula allows parties to show very efficiently that they follow the protocol.

We assume that parties S, R have once and for all set up their own instance of the commitment scheme and have proved in zero-knowledge that they know the corresponding secret keys, hence that a party can open any commitment w.r.t. his own public key.

COT Protocol

Input: Commitments A_0, A_1 by S, commitment B by R.

1. R makes a random commitment D, *using the public key of S*. If $\tilde{D} = 0$, let $T = DA_{\tilde{B}}$, else let $T = DA_{\tilde{B}}^{-1}$. Thus T is a random commitment, with distribution independent of $\tilde{B}$. T is sent to S.

2. R proves in zero-knowledge to S that T was correctly computed, i.e. proves that the following formula is satisfied:

$$(\tilde{B} = 0\ AND\ (\widetilde{TA_0^{-1}} = 0\ OR\ \widetilde{TA_0} = 1))$$

$$OR\ (\tilde{B} = 1\ AND\ (\widetilde{TA_1^{-1}} = 0\ OR\ \widetilde{TA_1} = 1))$$

Note that even though some of the commitments made are with S's public key, R knows enough about them to do the proof.

3. If the previous proof was accepted, S computes the bit contained in T and reveals it to R. Also, S proves that this bit is correct (S can compute only the bit, and not the random choices used for construction of T, so an interactive proof is necessary here).

4. R makes a commitment to C to the bit $\tilde{T} \oplus \tilde{D} = \tilde{A}_{\tilde{B}}$, and sends it to S. Proves also that C contains the correct bit, by showing that the formula

$$(\tilde{B} = 0\ AND\ \widetilde{TA_0^{-1}} = 0\ AND\ \tilde{C} = \tilde{T})$$

$$OR\ (\tilde{B} = 0\ AND\ \widetilde{TA_0} = 1\ AND\ \tilde{C} = 1 - \tilde{T})$$

$$OR\ (\tilde{B} = 1\ AND\ \widetilde{TA_1^{-1}} = 0\ AND\ \tilde{C} = \tilde{T})$$

$$OR\ (\tilde{B} = 1\ AND\ \widetilde{TA_1} = 1\ AND\ \tilde{C} = 1 - \tilde{T})$$

is satisfied.

To argue that this protocol has the required properties, note the following. The communication complexity follows since in each interactive proof, the verifier can commit to his challenge bit in one commitment as mentioned earlier. Hence proofs take a constant number of commitments, since the formulas have constant size.

Since T is distributed independently from $\tilde{B}$, all proofs are zero-knowledge, and commitments B and C are never opened, the protocol does not help S to compute $\tilde{B}$. Since the protocol can be simulated against any strategy by R knowing only $\tilde{A_{\tilde{B}}}$ (by using rewinding in step 2), the protocol does not help R to compute $\tilde{A_{1-\tilde{B}}}$.

It follows from soundness of the proof in step 4 that C contains the correct bit, except with negligible probability.

If this protocol is used as a building block in e.g. the construction of [7], then a multiparty computation protocol for evaluating a circuit of size n will take communication corresponding to $O(np^2)$ commitments of size k bits for an error probability of 2^{-k}. Earlier solutions, such as [6], typically require $\Omega(np^2k)$ commitments.

References

[1] L. Babai, L. Fortnow, L. Levin and M. Szegedi: *Checking Computations in Poly-logarithmic Time*, Proceedings of STOC '91.

[2] M. Bellare, S. Micali and R. Ostrovsky: *The (True) Complexity of Statistical Zero-Knowledge*, Proceedings of STOC '90, pp. 494–502.

[3] M. Bellare and O. Goldreich: *On Defining Proofs of Knowledge*, Proceedings of Crypto '92, Springer Verlag LNCS, vol. 740, pp. 390–420.

[4] J. Boyar, G. Brassard and R. Peralta: *Subquadratic Zero-Knowledge*, Journal of the ACM, November 1995.

[5] G. Brassard, D. Chaum and C. Crépeau: *Minimum Disclosure Proofs of Knowledge*, JCSS, vol.37, pp. 156–189, 1988.

[6] D. Chaum, I. Damgård and J. van de Graaf: *Multiparty Computations ensuring Privacy of each Party's Input and Correctness of the Result*, Proceedings of Crypto '87, Springer Verlag LNCS, vol. 293, pp. 87–119.

[7] C. Crépeau, J. van de Graaf and A. Tapp: *Comitted Oblivious Transfer and Private Multiparty Computation*, Proceedings of Crypto 95, Springer Verlag LNCS series, vol. 963.

[8] R. Cramer, I. Damgård and B. Schoenmakers: *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols*, Proceedings of Crypto '94, Springer verlag LNCS, vol. 839, pp. 174–187.

[9] R. Cramer: *Modular Design of Secure yet Practical Cryptographic Protocols*, Ph.D.-thesis, CWI & Uni. of Amsterdam, January 1997.

[10] I. Damgård, O. Goldreich, T. Okamoto and A. Wigderson: *Honest Verifier vs. Dishonest Verifier in Public Coin Zero Knowledge Proofs*, Proceedings of Crypto '95, Springer Verlag LNCS, vol. 963, pp. 325–338.

[11] I. Damgård and B. Pfitzmann: *On Soundness of Iterated Interactive Arguments and an Efficient Zero-Knowledge Argument for NP*, manuscript, February 1997.

[12] I.Damgård, T.P.Pedersen and B.Pfitzmann: *Statistical Secrecy and Multi-Bit Commitments*, BRICS report series RS-96-45, available at http://www.brics.dk. To appear in IEEE Trans.Info.Theory

[13] W. Diffie and M. Hellman: *New Directions in Cryptography*, IEEE Transactions on Information Theory IT-22 (6): 644–654, 1976.

[14] T. ElGamal, *A Public-Key Cryptosystem and a Signature Scheme based on Discrete Logarithms*, IEEE Transactions on Information Theory, IT-31 (4): 469–472, 1985.

[15] U. Feige and A. Shamir: *Witness Indistinguishable and Witness Hiding Protocols*, Proceedings of STOC '90, pp. 416–426.

[16] U. Feige and A. Shamir: *Zero-Knowledge Proofs of Knowledge in Two Rounds*, Proceedings of Crypto '89, Springer Verlag LNCS, vol. 435, pp. 526–544.

[17] U. Feige, A. Fiat and A. Shamir: *Zero-Knowledge Proofs of Identity*, Journal of Cryptology 1 (1988) 77–94.

[18] O. Goldreich, S. Micali and A. Wigderson: *Proofs that yield Nothing but their Validity and a Methodology of Cryptographic Protocol Design*, Proceedings of FOCS '86, pp. 174–187.

[19] O. Goldreich, S. Micali and A. Wigderson: *How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority*, Proceedings of STOC '87, ACM, pp. 218–229.

[20] O. Goldreich and A. Kahan: *How to Construct Constant-Round Zero-Knowledge Proof Systems for NP*, Journal of Cryptology, (1996) 9: 167–189.

[21] S. Goldwasser, S. Micali and C. Rackoff: *The Knowledge Complexity of Interactive Proof Systems*, SIAM J.Computing, Vol. 18, pp. 186-208, 1989.

[22] J. Kilian, S. Micali, and R. Ostrovsky: *Minimum resource zero-knowledge proofs*, Proceedings of FOCS '89, pp. 474–479.

[23] M. Naor: *Bit commitment using randomness*, Journal of Cryptology, (1991) 4 : 151–158.

[24] J. Kilian: *A note on Efficient Proofs and Arguments*, Proceedings of STOC '92.

[25] J. Kilian: *Efficient Interactive Arguments*, Proceedings of Crypto '95, Springer Verlag LNCS, vol. 963, pp. 311–324.

[26] T. Okamoto: *Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes*, Proceedings of Crypto '92, Springer Verlag LNCS, vol. 740, pp. 31–53.

[27] A. De Santis, G. Di Crescenzo, G. Persiano and M. Yung: *On Monotone Formula Closure of SZK*, Proceedings of FOCS '94, pp. 454–465.

[28] C. P. Schnorr: *Efficient Signature Generation by Smart Cards*, Journal of Cryptology, 4 (3): 161–174, 1991.

[29] A. Yao: *How to generate and exchange secrets*, Proceedings of FOCS '86, pp. 162–167.