European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2004
P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer (eds.)
Jyväskylä, 24–28 July 2004

STATISTICAL ZERO-KNOWLEDGE ARGUMENTS: THEORY AND PRACTICE

Helger Lipmaa

Laboratory for Theoretical CS, Department of CS&E
Helsinki University of Technology, P.O.Box 5400, FIN-02015 HUT, Espoo, Finland
email: helger@tcs.hut.fi, web page: http://www.tcs.hut.fi/~helger

Key words: Arguments of Knowledge, Diophantine Complexity, Electronic Auctions, Electronic Voting, Integer Commitment Scheme, Statistical Zero Knowledge.

Abstract. *During a statistical zero-knowledge argument, the arguer convinces the verifier on the truth of an assertment, without revealing next to nothing—but the truth of the assertment—even to an omnipotent verifier. The crucial part here is "next to nothing": compared to perfect zero-knowledge arguments where absolutely nothing (but the truth of the assertment) can be revealed, statistical zero-knowledge arguments are often much more practical.*

In this presentation, we will survey some recent developments in this area. We will both touch the theoretical and practical aspects of statistical zero-knowledge, ending the survey with my recent results that connect Hilbert's 10th problem to finding practical statistical zero-knowledge arguments.

1 INTRODUCTION

The notion of proof is central in mathematics. While it is fascinating to come up with a new hypothesis, the community will not usually accept it without a convincing and clear proof. It is required that the proof should be clear—in particular, easy to verify—and provide insights to why the claim happens to be true. In fact, the central open problem of theoretical computer science—$\mathcal{P} \stackrel{?}{=} \mathcal{NP}$—tackles precisely this question: show that in (some well defined sense) it is easier to verify a known proof than to prove new theorems.

On the other hand, it is not always desirable that the verifier will get any "insights" to why the claim is true. As an extreme example, assume that you have solved a millennium problem and want to claim 1,000,000 dollars. Clearly, for this you should be able to convince other people that you know the proof. But you also do not want to reveal the proof before you have received this money: for example, somebody might listen to your explanation of the proof and then present it as his or her own invention. Another example is from communication security: suppose that you want to convince that you are really you. A common way of doing that is to prove that you know your secret key, again, without revealing it.

One cannot achieve the described properties without first broadening the definition of a proof. To avoid non-transferability of the proof, it must both include randomness and interaction. One must also define what is meant by non-transferability of knowledge in this case. Thus, one arrives to the *zero-knowledge proofs*.

During a (perfect) zero-knowledge proof [15], the prover tries to convince the verifier in the truth of an claim so that with overwhelming probability, the verifier accepts the proof iff the claim is true. After the proof, the verifier should not be able to compute anything new that she was not able to compute before the zero-knowledge proof. This *zero-knowledge property* is formalized by using the simulation paradigm [15] by showing that the verifier can, without communicating with the prover, generate protocol views that have the same distribution as the real protocol views.

For the existence of zero-knowledge proofs one must assume that at least one of the two participants is computationally bounded. When the verifier is omnipotent but the prover is polynomially bounded—that is, a probabilistic polynomial time machine—one talks about *(perfect) zero-knowledge arguments* [6] (and not proofs) and about an arguer (not a prover). Since in practice, the verifier can often spend considerable amount of resources to break the zero-knowledge property after the protocol run but arguer can use his resources only during the protocol run, zero-knowledge arguments have a significant importance.

However, perfect zero-knowledge proofs and arguments are often inefficient for a practical use. Therefore, one would like to relax the security somewhat. A *statistical zero-knowledge (SZK) argument* is secure under the same conditions as a zero-knowledge argument, except that in the simulation, the simulator is only required to produce protocol views from a distribution that is statistically close to the distribution of real protocol

views. More precisely, it is required that the statistical difference between these two distributions is negligible in the length of the common input. Thus, during a single protocol run, the verifier obtains statistically insignificant amount of data about arguer's secret values; to obtain any significant information, the arguer and verifier must execute a super-polynomial number of protocol runs. The latter cannot happen since the arguer is polynomially bounded. Due to the relaxed definition, the SZK arguments are often much more practical than the (perfect) zero-knowledge arguments.

We survey some recent developments in the area of SZK arguments. We first show how much can be proven in SZK at all [26]. We then present our own recent results that connect Hilbert's 10th problem to finding practical SZK arguments [17]. We are able to construct very efficient SZK arguments for all languages in the bounded arithmetic. This improves in many aspects over previous work. Finally, we showing how to use the described results to construct efficient voting and auctions protocols. The latter part of the survey also includes some previously unpublished results and corrections to [17].

2 STATISTICAL ZERO-KNOWLEDGE ARGUMENTS: DEFINITION

We first proceed with definitions of statistical zero-knowledge (SZK) arguments. As noted before, such arguments are interactive—involving at least two participants, Arguer A and Verifier V—and randomized. We will model A and V as probabilistic polynomial-time computable functions. In a concrete protocol, at every step, such a function takes as arguments his or her private inputs, random coins and messages thus far, and produces the next message. At every moment, either party can halt the interaction (with acceptance, rejection, or no output). Formally,

Definition 1 (Interactive protocols) An *interactive protocol* (A, V) is a pair of functions. The interaction between A and V on common input x is the following random process, denoted as $(A, V)(x)$:

1. Uniformly choose random coins r_A and r_V (infinite binary strings) for A and V, respectively.
2. Repeat the following for $i = 1, 2, \ldots$:
   (a) If i is odd, let $m_i = A(x; m_1, \ldots, m_{i-1}; r_A)$.
   (b) If i is even, let $m_i = V(x; m_1, \ldots, m_{i-1}; r_v)$.
   (c) If m_i is one of ACCEPT, REJECT and HALT then stop. In this case, i is the number of the rounds of this protocol run.

The communication of a protocol run is the total number of bits exchanged. Protocol (A, V) is polynomially bounded if there exists a polynomial p, such that the sum of the number of the rounds and of the communication is at most $p(|x|)$, with probability 1 over the choice of r_A and r_V.

(As usually, we denote the length of a bit-string x by $|x|$.)

Definition 2 (Interactive argument system) Let (A, V) be an interactive protocol and let L be a language. We say that (A, V) is an *interactive argument system for* L with completeness error $c : \mathbb{N} \to [0, 1]$ and soundness error $s : \mathbb{N} \to [0, 1]$ if the following conditions hold:

1. (A, V) is polynomially bounded, A is polynomial-time computable.
2. If $x \in L$ then V accepts with probability at least $1 - c(|x|)$ in $(A, V)(x)$.
3. If $x \notin L$ then for any polynomially bounded A^*, V rejects with probability at least $1 - s(|x|)$ in $(A^*, V)(x)$.

$c(x)$ and $s(x)$ should be computable in polynomial time, and $1 - c(|x|) > s(|x|) + 1/p(|x|)$ for some polynomial p.

When V (and not A) is required to be polynomial-time computable then we get an *interactive proof system*. Both the completeness error and the soundness error can be reduced to $2^{|x|}$ by repeating the protocol polynomial number of times in $|x|$.

SZK arguments are interactive argument systems with an additional zero-knowledge property that is defined by using the simulatability paradigm. To define this, we first must define what is a protocol view.

Definition 3 (Protocol view) Let (A, V) be an interactive protocol. V's view of (A, V) on common input x is the random variable $\langle A, V\rangle(x) = (m_1, \ldots, m_R; r)$ that contains all messages m_i exchanged during a protocol view and the random string r that contains all bits of r_V that were actually read during the interaction. (A's view is defined dually. However, we will only be interested in V's view.)

We also need the notion of statistical difference.

Definition 4 (Statistical difference) If X and Y are probability distributions on a discrete universe U, then the statistical difference between X and Y is defined as

$$\mathsf{StatDiff}(X, Y) := \max_{S \subseteq U} |\Pr[X \in S] - \Pr[Y \in S]| \ .$$

By definition, $\mathsf{StatDiff}(X, Y) \in [0, 1]$, $\mathsf{StatDiff}(X, Y) = 0$ if X and Y are identical, $\mathsf{StatDiff}(X, Y) = 1$ if X and Y have disjoint supports and that $\mathsf{StatDiff}(X, Z) \leq \mathsf{StatDiff}(X, Y) + \mathsf{StatDiff}(Y, Z)$.

Now we are almost ready to define what is an SZK argument. Recall first the situation. Arguer A wants to convince verifier V that $x \in L$, where x is the common input of A and V, and L is some language. For example, x could be a large integer and A wants to convince V that x is a prime. On the other hand, A does not want V to be able to transfer

this knowledge to anybody else. In particular, A does not want to reveal to V any new information that V did not already know. What does it mean for V not to gain any new information is best formalized by using the notion of a simulator. Namely, V will not gain any new information from interaction with A when there exists a probabilistic polynomial time (PPT) machine S that "simulates" V's interaction with A: that is, produces output that is identical or very close to $\langle A, V\rangle(x)$. Now, since V could have executed S himself to produce the protocol view that he obtains from interaction with A, clearly he does not obtain any new information.

Next, recall that we are interested in statistical zero-knowledge. That is, not in the case where S's output and $\langle A, V\rangle(x)$ are identically distributed but just in the case when these two distributions are statistically close. For a more precise definition, recall that A is polynomially bounded. In particular, this means that the protocol (A, V) can be executed only a polynomial number of times. We want the statistical difference between the two distributions to be small enough so that the total leakage of the information during a polynomial number of rounds in $|x|$ is still insignificant. This is formalized by requiring that the statistical difference must be negligible. (Recall that a function f is negligible when for any polynomial p there exists a k_0 such that for all $k > k_0$, $f(k) < p(k)$.)

For technical reasons, it is also necessary that the simulator can fail with probability $< 1/2$, by outputing FAIL. Thus, the quality of the simulation will be measured conditioned on the non-failure.

Finally, during most of this paper we will only need *honest-verifier* SZK where the simulator is required to simulate (A, V) only when V follows the protocol. Therefore, we will first give the definition for the honest-verifier case.

Definition 5 (Honest-Verifier SZK (HVSZK) Argument) An interactive argument system (A, V) for a language L is said to be *honest-verifier statistical zero-knowledge* if there is a PPT algorithm S (that fails with probability $< 1/2$) and a negligible function f, such that for all $x \in L$ and all $k > 0$,

$$\mathsf{StatDiff}(S(V, x), \langle A, V\rangle(x)) \leq f(|x|) \ .$$

f is called the simulator deviation. If $f \equiv 0$ then (A, V) is a honest-verifier perfect zero-knowledge argument. $\mathcal{HVSZK}^{\mathcal{A}}$ denotes the class of languages that have HVSZK arguments.

When V (and not A) is required to be polynomial-time computable then we get an *honest-verifier SZK proof*. In this case, one gets a *honest-verifier computational zero-knowledge (HVCZK) proof* when instead of requiring that the distributions $S(V, x)$ and $\langle A, V\rangle(x)$ are statistically close, one just requires that these distributions are computationally indistinguishable, that is, that there does not exist a probabilistic polynomial-time machine M that can distinguish the distributions with non-negligible advantage. Zero-knowledge arguments are also sometimes known as *computationally-convincing or computationally-*

sound zero-knowledge proofs. $\mathcal{HVSZK}^{\mathcal{P}}$ denotes the class of languages that have HVSZK proofs.

Definition 6 (SZK Argument) An interactive argument system (A, V) for a language L is said to be *statistical zero-knowledge* if there is a PPT algorithm S (that fails with probability $< 1/2$) and a negligible function f, such that for all $x \in L$, all PPT machines V^*, and all $k > 0$,

$$\mathsf{StatDiff}(S(V^*, x), \langle A, V^* \rangle(x)) \leq f(|x|) \ .$$

f is called the simulator deviation for V^*. If $f \equiv 0$ then (A, V) is a perfect zero-knowledge argument. $\mathcal{SZK}^{\mathcal{A}}$ denotes the class of languages that have SZK arguments.

Finally, note that in the definition of zero-knowledge, technical details matter a lot. Since our emphasis is on Section 4—that is, on efficient (honest-verifier) SZK arguments—and we do not intend to present a complete survey, we will omit such details.

3 ON THE POWER OF SZK ARGUMENTS

3.1 Complexity of $\mathcal{HVSZK}^{\mathcal{P}}$

Let us first look at the better studied class $\mathcal{HVSZK}^{\mathcal{P}}$. The structure of $\mathcal{HVSZK}^{\mathcal{P}}$ has been studied quite thoroughly [26] and several fundamental results are known about it. (We will only cite [26], the original references as well as undefined notions can be obtained from there.) For example, several complete problems are known for this class.

Definition 7 (Promise problem STATDIFF) Assume that two Boolean circuits encode two distributions X and Y. On the promise that either $\mathsf{StatDiff}(X, Y) \geq 2/3$ or $\mathsf{StatDiff}(X, Y) \leq 1/3$, decide which one is the case.

Theorem 1 *STATDIFF is $\mathcal{HVSZK}^{\mathcal{P}}$-complete.*

From the completeness of STATDIFF—and more precisely, from the concrete HVSZK protocol for it—and from several important other results it follows that

Theorem 2 *If $\mathcal{NP} \subseteq \mathcal{HVSZK}^{\mathcal{P}}$ then the Polynomial Hierarchy collapses.*

Therefore, given the current state of the knowledge on the Polynomial Hierarchy, it is not believed that $\mathcal{NP} \subseteq \mathcal{HVSZK}^{\mathcal{P}}$. Finally, it is known that

Theorem 3 $\mathcal{SZK}^{\mathcal{P}} = \mathcal{HVSZK}^{\mathcal{P}}$.

3.2 Complexity of $\mathcal{HVSZK}^{\mathcal{A}}$

Compared to the HVSZK proofs, in the HVSZK arguments verifiers lose something in security—namely, their security is only guaranteed in the case when the prover is computationally bounded. However, as argued before, this is not a great loss since the

prover must be able break the protocol "on the spot"—that is, she must be able to convince the verifier in the truth of a false statement within a few seconds. On the other hand, the verifier would have all the time in the universe to break the zero-knowledge property of the protocol. Therefore, HVSZK arguments seem to be more suitable in the practice than HVSZK proofs—especially since in different applications (like electronic voting or polling on sensitive issues), provers might be very interested in achieving unconditional privacy (e.g., of their vote).

However, not that much is known about the SZK arguments than about the SZK proofs. Vadhan even stated in [26] that it may lead to a big research program to establish similar results about the SZK arguments as are known about the SZK proofs.

It also comes out that most probably, $\mathcal{SZK}^{\mathcal{P}} \subseteq \mathcal{SZK}^{\mathcal{A}}$. This is due to the fact that $\mathcal{NP} \subseteq \mathcal{SZK}^{\mathcal{A}} = \mathcal{HVSZK}^{\mathcal{A}}$. Let us look closer at the last claim. Most of the next results were proven for perfect zero-knowledge arguments and thus by extension also hold for statistical zero-knowledge arguments. First, $\mathcal{NP}$ has four-round computational zero-knowledge arguments assuming that one-way functions exist, as shown by Bellare, Jakobsson and Yung [4]. Both one-way functions and four rounds are also necessary for zero-knowledge proofs or arguments.

On the other hand, the known statistical zero-knowledge arguments for $\mathcal{NP}$ are both less efficient and are based on stricter assumptions. It was proven by Brassard, Chaum and Crépeau that $\mathcal{NP}$ has $\omega(\log n)$-round perfect zero-knowledge arguments, based on an algebraic assumption [6]. After that, Brassard, Crépeau and Yung proved that $\mathcal{NP}$ has 6-round perfect-zero-knowledge arguments, based on the existence of claw-free pairs [7]. Finally, Naor, Ostrovsky, Venkatesan and Yung showed that poly(n)-round arguments for $\mathcal{NP}$ exist only on the assumption of the existence of one-way permutations.

From the negative side, Goldreich and Krawczyk proved that NP does not have three-round black box zero-knowledge arguments [14]. In general, in zero-knowledge arguments and proofs the simulator has the big disadvantage compared to the actual (possibly cheating) verifier V^* that he does not necessarily know the proved object. To make up for it, the simulator is given access to the random tape of V^*, together with the ability to rewind V^*, that is, to re-execute V^* by using the same coins. In the case of a black box zero-knowledge argument, the simulator has only the oracle access to the verifier—and no access to the verifier's code—and thus cannot make any use of the knowledge of the random tape—for example, since the random tape can also be partially coded in the verifier's code. Therefore, the simulator has only one advantage compared the prover: he can rewind the cheating prover.

All previous arguments used black-box simulators, that is, simulators that only have oracle access to the verifier. In 2001, Barak showed that a lot more can be achieved with non-black-box simulators. In particular, he proposed a new (perfect) zero-knowledge argument system for $\mathcal{NP}$ that has a constant number of rounds with negligible soundness error, uses only public coins (i.e., at every step, the verifier only sends randomly chosen bits), remains zero-knowledge even when composed concurrently a relatively large number

of times, and has a simulator that runs in strict polynomial time [2]. Several of these properties were known to be contradictory in the case of black-box simulators. Moreover, as later shown by Barak and Goldreich [3], such arguments exist when just assuming the existence of collision-resistant hash-functions. Shortly, Barak's idea was to use the description of the verifier's next-round function as an additional trapdoor information.

4 EFFICIENT SZK ARGUMENTS FOR BOUNDED ARITHMETIC

In the previous section, we gave a short overview of the theoretical limits of the statistical zero-knowledge arguments. In this section, we will concentrate on the practical aspects.

The correctness of cryptographic protocols is usually guaranteed by accompanying every step of the protocol with a zero-knowledge proof or argument of "well-behavedness". Like already mentioned, (statistical) ZK arguments are often more suitable in practice since they guarantee statistical privacy to the prover. In different protocols (like voting or polling on sensitive issues) the prover (the voter or the respondent) might be reluctant to participate without such a strong guarantee on his or her privacy.

When all steps of a protocol are accompanied with an SZK argument, one must take special care not to make the resulting protocol too inefficient. For example, a quadratic "zero-knowledge" overhead in communication is too much for most of the application. From the theoretical viewpoint, quadratic is still quadratic, but in practice this might mean a 1000-fold increase in communication! This would mean that using secure protocols would be too resource-consuming and different parties (companies, governments and private citizens) would continue using insecure versions. Therefore, it is extremely important to construct extremely efficient SZK arguments for the practical protocols like electronic voting, auctions, polling and cash.

4.1 Cryptographic Preliminaries

There exist several methodologies to design efficient cryptographic protocols. The one that is most used nowadays (and often also results in the best protocols) is based on homomorphic encryption.

Recall that public-key cryptosystem is a triple $\Pi = (G, E, D)$ where G is the key generation algorithm, E is the encryption algorithm and D is the decryption algorithm. Denote the encryption of message m as $E_K(m; r)$ where K is the used public key and r is the used random coin. Assume that the ciphertext space (resp. plaintext space) is a multiplicative (resp. additive) group with group operation $\cdot$ (resp. $+$). Assume the random elements are drawn from some groupoid with groupoid operation $\circ$. Π is *homomorphic* when $E_K(m_1; r_1) \cdot E_K(m_2; r_2) = E_K(m_1 + m_2; r_1 \circ r_2)$ for any valid public key K, messages m_i and random coins r_i. Paillier's cryptosystem [23] is one of the well-known homomorphic cryptosystems.

4.1.1 Electronic Voting.

In electronic voting, the ith voter submits an encrypted vote v_i. It is required that the voting center gets to know how many voters voted for every candidate, but not how did every single voter vote. This can be done as follows by using a homomorphic cryptosystem [8, 11]: Let a be the upper limit to the number of voters. There is also $3t+1$ servers that share a public key K and a private key x so that everybody can encrypt a message by using K, but only $2t+1$ servers can jointly decrypt the ciphertext. The ith voter encrypts a^{v_i} by using the key K and sends it to the servers. The servers collect all ciphertexts and return receipts to the voters. After the end of the election, the servers multiply all ciphertexts, getting $y = E_K(\sum_i a^{v_i}) = E_K(\sum \alpha_j a^j)$, where α_j is the number of voters who voted for the candidate j. Thus, the servers can jointly decrypt y, and then compute the coefficients α_j.

To guarantee the correctness of this protocol, all voters must prove or argue in honest-verifier ZK that they encrypted a value of form a^j where $j \in [0, m-1]$ where m is the number of candidates. This is called a *range argument in exponents* (RAIE). Damgård and Jurik [11] proposed a computational RAIE. As improved by Lipmaa, Asokan and Niemi [19], this honest-verifier computational ZK (HVCZK) proof has communication $\Theta(\max(k, m \log a) \cdot \log m) = \Theta(m \cdot \log a \cdot \log m)$. The latter proof seems to be the most efficient known computational honest-verifier ZK proof for RAIE.

4.1.2 Electronic Vickrey Auctions.

In Vickrey auctions, the ith bidder submits an encrypted bid v_i. The seller should get to know the highest bidder and the second highest bid but nothing else. Also this can be done by using a homomorphic cryptosystem as follows [19]. Assume this time that there is a single seller S and an established auction authority A; so that one of the two may be malicious but they do not collaborate. Let a be the fixed maximum number of bidders. The ith bidder encrypts a^{b_i} by using A's public key K and sends the result to S. S collects the ciphertexts, sends back receipts. After that, he multiplies the ciphertexts and gets $y = E_K(\sum \alpha_j a^j)$, where α_j is the number of bidders who bid j. S sends y to A. A decrypts the result and sends the highest bid X_1 (in encrypted form). As suggested in [18], after that the bidders, S and A will participate in proxy private equality test, by proving (without knowing) that their bids were or were not equal to X_1. After the end of private equality tests, A helps S to reveal which bidders bid X_1, and also sends X_2 to him.

To guarantee the correctness, all bidders must again do a RAIE. Additionally, A must prove that X_2 and X_1 were correctly computed. As shown in [19], this means that A must do an additional RAIE, but also a range argument (RA) that some encrypted value belongs to the range $[L, H]$. For the latter, the known HVCZK proofs are approximately as efficient as for the RAIE [5]. In particular, when HVCZK proofs are used, the Vickrey auction scheme from [19] is not significantly more efficient than the scheme of Naor, Pinkas

and Sumner [22] that is based on the general two-party computation.

To make their protocols more efficient, [19] proposed to use HVSZK arguments instead of the HVCZK proofs. In the next we will describe the methodology by Lipmaa [17] that was used in [19] for this purpose but can also be used to solve other problems. This methodology bases on two pillars, cryptography and Diophantine complexity.

4.2 Integer Commitment Schemes

In 1997, Fujisaki and Okamoto introduced a new primitive called an *integer commitment scheme*. Recall that a commitment scheme is a function $C : X \times R \to Y$ from the plaintext space X and random coin space R to the commitment space Y. A commitment scheme C is said to be (a) statistically hiding if the commitment y $= C(x; r)$ leaks a statistically insignificant amount of information about the plaintext x and the coin r; and (b) computationally binding if given commitment $y = C(x; r)$ to some element r from the plaintext space, it is hard to find $x' \neq x$ from the plaintext space and r', s.t. $y = C(x'; r')$. For the best known commitment schemes, the plaintext space is equal to $\mathbb{Z}_n$ for some n. Therefore, it is always the case that $C(x; r) = C(x+n; r)$ and therefore, such commitment schemes are not binding over the integers.

Fujisaki and Okamoto designed a new integer commitment scheme [12] with the property that for any $x \in \mathbb{Z}$ and random coin r, it is hard to find $x' \neq x$ (as an integer) and an r', s.t. $C(x; r) = C(x'; r')$. Later, this commitment scheme was improved by Damgård and Fujisaki [9]. Note that both commitment schemes are computationally binding and statistically hiding.

Both the Fujisaki-Okamoto [12] and the Damgård-Fujisaki [9] integer commitment schemes are homomorphic, with $C(x; r) \cdot C(x'; r') = C(x + x'; r + r')$. Moreover, for both schemes one can construct an extremely efficient HVSZK argument of knowledge that given three commitments c_1, c_2 and c_3, the prover knows such x_1 and x_2 and corresponding random coins r_1, r_2 and r_3, that $c_1 = C(x_1; r_1)$, $c_2 = C(x_2; r_2)$ and $c_3 = C(x_1 x_2; r_3)$. As already emphasis ed in [12], one can use the homomorphic property of integer commitment schemes together with the efficient HVSZK argument of knowledge for the multiplicative relation to construct efficient HVSZK arguments of knowledge of type $c_1 = C(x_1; r_1)$, $\ldots$, $c_n = C(x_n; r_n)$, $c_{n+1} = C(p(x_1, \ldots, x_n); r_{n+1})$, where p is an arbitrary polynomial $p \in \mathbb{Z}[X_1, \ldots, X_n]$.

This enables one to construct efficient HVSZK arguments of knowledge for several cryptographically interesting relations. For example, as shown by Boudot, there exists a HVSZK argument of knowledge for the range argument $y = C(x; r) \wedge x \in [L, H]$, that is linearly long. However, Boudot's argument is somewhat intuitive and does not seem to generalize to other interesting relations.

4.3 Diophantine Complexity

Lipmaa's methodology for constructing HVSZK arguments of knowledge is intimately connected with the Diophantine complexity that, in its turn, is connected to the fundamental work of Davis, Putnam, Robinson and Matiyasevich in 1950–1990. Based on the earlier work of Davis, Putnam and Robinson, Matiyasevich proved in 1970 [20] that every recursively enumerable set is Diophantine (this important result is known as the DPRM theorem), solving thus negatively Hilbert's tenth problem from year 1900.

Recall that a set $S \subset \mathbb{Z}^n$ is called *Diophantine* [21], if it has a *representing polynomial* $\mathfrak{R}_S \in \mathbb{Z}[X;Y]$, $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_m)$, such that $\mu \in S$ iff for some witness $\omega \in \mathbb{Z}^m$, $\mathfrak{R}_S(\mu;\omega) = 0$. This work on the Hilbert's tenth problem has had many interesting consequences. In particular, it means that there exists a universal polynomial U. See [21] for a representation of main results of this work and related history.

In 1976, Adleman and Manders [1] introduced a new complexity-theoretic class $\mathcal{D}$ of sets, defined as follows: $S \in \mathcal{D}$ iff there exists a representing polynomial $\mathfrak{R}_S$, such that

$$\mu \in S \iff (\exists \omega)\Big[\Big|\sum_i \omega_j\Big| = \Big|\sum_i \mu_i\Big|^{O(1)} \wedge \mathfrak{R}_S(\mu;\omega) = 0\Big] \ .$$

Compared to the definition of Diophantine sets, here it is allowed that there exist superpolynomially long witnesses in the case $\mu \notin S$.

Obviously, $\mathcal{D} \subseteq \mathcal{NP}$. On the other hand, Adleman and Manders showed that several $\mathcal{NP}$-complete problems belong to the class $\mathcal{D}$ and, based on that, conjectured that $\mathcal{D} = \mathcal{NP}$. Their conjecture was later implicitly supported by Jones and Matiyasevich [16] who proved that $\mathcal{D} = \mathcal{NP}$ iff the set $\{(\mu_1, \mu_2) : \mu_1 \leq_2 \mu_2\}$ belongs to $\mathcal{D}$ (here, $\mu_1 \leq_2 \mu_2$ iff $\mu_1 \wedge \mu_2 = \mu_1$, where $\wedge$ denotes the bitwise and) and by Pollett [24], who showed that if the $\mathcal{FDLOGTIME}$ functions are definable in $\mathcal{D}$ then $\mathcal{D} = \mathcal{NP}$, where $\mathcal{FDLOGTIME}$ is a well-known relatively small complexity class. (A function is in $\mathcal{FDLOGTIME}$ when it has a polynomial-sized bit-graph whose bits are in $\mathcal{DLOGTIME}$.) The gap between $\mathcal{FDLOGTIME}$ and $\mathcal{NP}$ is wide and thus, as expected, not much is known about the actual power of the class $\mathcal{D}$.

Lipmaa [17] introduces a new complexity class $\mathcal{PD}$ that is a Diophantine analogue of $\mathcal{P}$. Namely, $S \in \mathcal{PD}$ iff there is a polynomial $\mathfrak{R}_S \in \mathbb{Z}[X]$, such that (1) there exists a probabilistic polynomial-time *witness algorithm* $\mathfrak{P}_S$, such that if $\mu \in S$ then $\mathfrak{R}_S(\mu; \mathfrak{P}_S(\mu)) = 0$; (2) if $\mu \notin S$ then for any ω with $|\omega| = |\mu|^{O(1)}$, $\mathfrak{R}_S(\mu;\omega) \neq 0$.

So how large is $\mathcal{PD}$? The answer is that we do not know. Clearly, $\mathcal{PD} \subseteq \mathcal{D}$. Moreover, if factoring is hard then $\mathcal{PD} \neq \mathcal{D}$: the language of composite integers belongs to $\mathcal{D}$, but to find the witnesses one needs to factor large integers.

On the other hand, as shown in [17], all languages in bounded arithmetic L_2 belong to $\mathcal{PD}$. Recall that bounded arithmetic is a first-order theory of the natural numbers with non-logical symbols 0, σ, $+$, $\cdot$, $\leq$, $\dot{-}$, $\lfloor x/2 \rfloor$, $|x|$, $\mathsf{MSP}(x,i)$ and $\sharp$. The symbols 0, $\sigma(x) := x + 1$, $+$, $\cdot$, and $\leq$ have their usual meaning. Other operations are defined as

$x \dot{-} y := \max(x - y, 0)$, $|x| := \lfloor \log_2(x+1) \rfloor$, $\mathsf{MSP}(x,i) := \lfloor x/2^i \rfloor$ and $x \sharp y := 2^{|x| \cdot |y|}$. (For our purposes we adapt a slightly modified definition of bounded arithmetic where the underlying domain is $\mathbb{Z}$ instead of $\mathbb{N}$.) We denote by L_2 the set of terms of the quantifier-free bounded arithmetic (over $\mathbb{Z}$).

Moreover, [17] shows that most of the primitive operations of bounded arithmetic have representing polynomials with linear-size witnesses. The only exception is $x \sharp y$ with super-linear (but sub-quadratic) witnesses. The intuition is that one can find a representing polynomial for $x \sharp y$ easily by using a representing polynomial for the exponential relation $c = a^b$, and the latter polynomial has sub-quadratic (but super-linear) witnesses.

The proof that $L_2 \subseteq \mathcal{PD}$ contains also concrete nontrivial representing polynomials for some of the relations. The first representing polynomial is for the relation $x \geq 0$; it bases on the classical result of Lagrange from 1770 that every nonnegative integer is a sum of four squares and on the relatively recent result of Rabin and Shallit [25] that these four squares can be computed efficiently. ([17] proposes a slightly more efficient algorithm for the same task.)

4.3.1 Cryptographic Applications.

Given a secure integer commitment scheme with efficient HVSZK arguments of knowledge for additive and multiplicative relations, one can argue in HVSZK that any polynomial relation holds between a tuple of committed integers [13]. That is, one can argue in HVSZK that $p(\mu) = 0$ for some fixed $p \in \mathbb{Z}[X]$, and a committed $\mu \in \mathbb{Z}^n$. Lipmaa expanded the [13]-methodology as follows. When $S \in \mathcal{D}$ and the arguer knows the witness, then by using an integer commitment scheme, she can argue in HVSZK that she knows an auxiliary (suitably chosen) witness ω, such that $\mathfrak{R}_S(\mu; \omega) = 0$, where $\mathfrak{R}_S$ is again the representing polynomial of S. This results in an efficient *Diophantine argument system*.

So, how can one use this theory in practice? When combined with the integer commitment schemes, this means that every language in bounded arithmetic has an HVSZK argument of knowledge with subquadratic length. Due to the Lagrange's polynomial and the Rabin-Shallit-Lipmaa algorithm, the range argument—that was very useful in the previously described Vickrey auction scheme—has a HVSZK argument with linear length. RAIE—used in both voting and auctions—has also subquadratic length. However, a subquadratic RAIE is not sufficient since the HVCZK proof of knowledge, proposed in [11, 19], is more efficient.

Due to that, [17] proposed to use another function $a^{[\![n]\!]}$ instead of the exponentiation a^n. Recall that all nonnegative integral solutions (x, y) of the equation $x^2 - axy - y^2 = 1$ are either equal to $(a^{[\![n+1]\!]}, a^{[\![n]\!]})$ or $(a^{[\![n]\!]}, a^{[\![n+1]\!]})$, $n \geq 0$, where $a^{[\![n]\!]}$ can be computed by using the next recurrent identities [21]: $a^{[\![0]\!]} := 0$, $a^{[\![1]\!]} := 1$, and $a^{[\![n+2]\!]} := aa^{[\![n+1]\!]} - a^{[\![n]\!]}$ for $n \geq 0$. Thus, $\{a^{[\![n]\!]}\}_{n \in \mathbb{N}}$ is a Lucas sequence. When $a > 2$ and $n > 0$ then $(a-1)^n \leq a^{[\![n+1]\!]} \leq a^n$. Also, $a^{[\![n]\!]}$ can be computed almost as efficiently as a^n. Therefore, $a^{[\![n]\!]}$ is exactly as suitable to use as the encoding function that the voters used in the Damgård-

Jurik voting scheme (the same applies in the Lipmaa-Asokan-Niemi auction scheme): instead, of sending $E_K(B^{v_i}; r)$ to the talliers, the voters can send $E_K((B+1)^{[\![v_i]\!]}; r)$. Since $(a-1)^n \leq a^{[\![n+1]\!]} \leq a^n$, then the talliers can still recover the coefficients α_j from the sum $\sum_i (B+1)^{[\![v_i]\!]}$.

However, due to the fact that $(a^{[\![n]\!]})^2 - aa^{[\![n]\!]}a^{[\![n+1]\!]} - (a^{[\![n+1]\!]})^2 = 1$, there is a $\Theta(m \log a)$-bit HVSZK argument of knowledge to prove that a voter voted correctly, where a is the number of voters and m is the number of possible choices (i.e., the number of candidates). This is $\Theta(\log m)$ times more efficient than the HVCZK proof of knowledge from [11, 19]. The RAIE is the single most communication-consuming subprotocol of both the Damgård-Jurik voting scheme and of the Lipmaa-Asokan-Niemi auction scheme. Therefore, the use of $(a+1)^{[\![n]\!]}$ instead of a^n in both results in $\Theta(\log m)$-fold decrease of total communication in both schemes.

Note that Lipmaa, Asokan and Niemi [19] proposed an alternative RAIE that is based on the methodology from [17]. Instead of the function a^n (or $a^{[\![n]\!]}$, it uses the function b^n, where b is the least prime greater than equal to a. Since a is fixed a priori and publicly known, b can computed before the electronic voting or auction starts. This RAIE is approximately as efficient as the RAIE based on the Lucas sequences: the arguer must argue that the committed value μ is such that $b^L \mid \mu$ and $\mu \mid b^H$. As later shown in [10], one can simplify the argument even more by assuming that $b = p^2$ for a prime p.

4.3.2 Outsourcing Protocols.

Recall that [19] made use of the auction authority A. The seller S sends some commitment $y = C(m; r)$ to A, A computes some function $f(m)$, and sends it—together with a HVSZK argument of knowledge—back to S. Now, A gets to know the full value of m (in this case, how the bids are distributed) but is unable to connect this information with the concrete bidders. S, on the other hand, will obtain only $f(m)$ (e.g., the second highest bid). Since anybody can be the seller, while such an authority A would have a long reputation history, revealing m to A seems to be relatively harmless.

The same methodology can be used in an arbitrary protocol where the function f is in bounded arithmetic. This is the case in auctions (including the $(m+1)$st-price auctions), voting, etc. Indeed, it seems to be the case in most of the widely-known cryptographic scenarios where some set of participants have to make some social and financial choices and a center must compute an outcome that is based on such choices. Thus, in all such scenarios where $f \in L_2$, one can do with sub-quadratic-size total communication. In every scenario that we are aware of, the communication is actually linear. This includes some cryptographic tasks for which no efficient solution was previously known at all.

Acknowledgements

We would like to thank Kaisa Nyberg for helpful comments. This work was partially supported by the Finnish Defence Forces Technical Research Centre.

REFERENCES

[1] Leonard M. Adleman and Kenneth L. Manders. Diophantine Complexity. In *17th Annual Symposium on Foundations of Computer Science*, pages 81–88, Houston, Texas, USA, 25–27 October 1976. IEEE Computer Society Press.

[2] Boaz Barak. How to Go Beyond the Black-Box Simulation Barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, Las Vegas, Nevada, USA, 14–17 October 2001. IEEE, IEEE Computer Society Press.

[3] Boaz Barak and Oded Goldreich. Universal Arguments and their Applications. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity*, pages 194–203, Montréal, Québec, Canada, 21–24 May 2002. IEEE Computer Society.

[4] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-Optimal Zero-Knowledge Arguments Based on any One-Way Function. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 280–305, Konstanz, Germany, 11–15 May 1997. Springer-Verlag.

[5] Fabrice Boudot. Efficient Proofs that a Committed Number Lies in an Interval. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444, Bruges, Belgium, 14–18 May 2000. Springer-Verlag.

[6] Guilles Brassard, David Chaum, and Claude Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[7] Guilles Brassard, Claude Crépeau, and Moti Yung. Constant Round Perfect Zero Knowledge Computationally Convincing Protocols. *Theoretical Computer Science*, 84(1):23–52, 1991.

[8] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118, Konstanz, Germany, 11–15 May 1997. Springer-Verlag.

[9] Ivan Damgård and Eiichiro Fujisaki. An Integer Commitment Scheme Based on Groups with Hidden Order. In Yuliang Zheng, editor, *Advances on Cryptology — ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142, Queenstown, New Zealand, 1–5 December 2002. Springer-Verlag.

[10] Ivan Damgård, Jens Groth, and Gorm Salomonsen. *The Theory and Implementation of an Electronic Voting System*, pages 77–99. Kluwer Academic Publishers, 2002.

[11] Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *Public Key Cryptography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, Korea, 13–15 February 2001. Springer-Verlag.

[12] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In Jr. Burton S. Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30, Santa Barbara, USA, 17–21 August 1997. Springer-Verlag. ISBN 3-540-63384-7.

[13] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical Zero-Knowledge Protocols to Prove Modular Polynomial Relations. *IEICE Transaction of Fundamentals of Electronic Communications and Computer Science*, E82-A(1):81–92, January 1999.

[14] Oded Goldreich and Hugo Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal of Computing*, 25(1):169–192, 1996.

[15] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal of Computing*, 18(1):186–208, 1989.

[16] J. P. Jones and Yuri Matiyasevich. Register Machine Proof of the Theorem on Exponential Diophantine Representation of Enumerable Sets. *Journal of Symbolic Logic*, 49:818–829, 1984.

[17] Helger Lipmaa. On Diophantine Complexity and Statistical Zero-Knowledge Arguments. In Chi Sung Laih, editor, *Advances on Cryptology — ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415, Taipei, Taiwan, 30 November–4 December 2003. Springer-Verlag.

[18] Helger Lipmaa. Verifiable Homomorphic Oblivious Transfer and Private Equality Test. In Chi Sung Laih, editor, *Advances on Cryptology — ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 416–433, Taipei, Taiwan, 30 November–4 December 2003. Springer-Verlag.

[19] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey Auctions without Threshold Trust. In Matt Blaze, editor, *Financial Cryptography — Sixth International Conference*, volume 2357 of *Lecture Notes in Computer Science*, pages 87–101, Southhampton Beach, Bermuda, 11–14 March 2002. Springer-Verlag.

[20] Yuri Matiyasevich. Enumerable Sets are Diophantine. *Soviet Math., Doklady*, 11:354–358, 1970. English translation.

[21] Yuri Matiyasevich. *Hilbert's Tenth Problem*. Foundations of Computing. MIT Press, October 1993. ISBN 0-262-13295-8.

[22] Moni Naor, Benny Pinkas, and Reuben Sumner. Privacy Preserving Auctions and Mechanism Design. In *The 1st ACM Conference on Electronic Commerce*, Denver, Colorado, November 1999.

[23] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, 2–6 May 1999. Springer-Verlag.

[24] Chris Pollett. On the Bounded Version of Hilbert's Tenth Problem. *Archive for Mathematical Logic*, 42(5):469–488, 2003.

[25] Michael O. Rabin and Jeffrey O. Shallit. Randomized Algorithms in Number Theory. *Communications in Pure and Applied Mathematics*, 39:239–256, 1986.

[26] Salil Pravin Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, MIT, USA, August 1999. Available from http://www.eecs.harvard.edu/~salil/papers/phdthesis-abs.html, as of March 2004.