On the Amortized Complexity of Zero Knowledge Protocols for Multiplicative Relations

Ronald Cramer, Ivan Damgård, and Valerio Pastro

CWI Amsterdam and Dept. of Computer Science, Aarhus University

Abstract. We present a protocol that allows to prove in zero-knowledge that committed values $x_i, y_i, z_i, i = 1, \ldots, l$ satisfy $x_i y_i = z_i$, where the values are taken from a finite field. For error probability $2^{-u}$ the size of the proof is linear in $u$ and only logarithmic in $l$. Therefore, for any fixed error probability, the amortized complexity vanishes as we increase $l$. In particular, when the committed values are from a field of small constant size, we improve complexity of previous solutions by a factor of $l$. Assuming preprocessing, we can make the commitments (and hence the protocol itself) be information theoretically secure. Using this type of commitments we obtain, in the preprocessing model, a perfect zero-knowledge interactive proof for circuit satisfiability of circuit $C$ where the proof has size $O(|C|)$. We then generalize our basic scheme to a protocol that verifies $l$ instances of an algebraic circuit $D$ over $K$ with $v$ inputs, in the following sense: given committed values $x_{i,j}$ and $z_i$, with $i = 1, \ldots, l$ and $j = 1, \ldots, v$, the prover shows that $D(x_{i,1}, \ldots, x_{i,v}) = z_i$ for $i = 1, \ldots, l$. The interesting property is that the amortized complexity of verifying one circuit only depends on the multiplicative depth of the circuit and not the size. So for circuits with small multiplicative depth, the amortized cost can be asymptotically smaller than the number of multiplications in $D$. Finally we look at commitments to integers, and we show how to implement information theoretically secure homomorphic commitments to integer values, based on preprocessing. After preprocessing, they require only a constant number of multiplications per commitment. We also show a variant of our basic protocol, which can verify $l$ integer multiplications with low amortized complexity. This protocol also works for standard computationally secure commitments and in this case we improve on security: whereas previous solutions with similar efficiency require the strong RSA assumption, we only need the assumption required by the commitment scheme itself, namely factoring.

1 Introduction

The notions of commitment schemes and zero-knowledge proofs are among the most fundamental in the theory and practice of cryptographic protocols. Intuitively, a commitment scheme provides a way for a prover to put a value $x$ in a locked box and commit to $x$ by giving this box $[x]$ to a verifier. Later the prover can choose to open the box by giving away the key to the box.
In a zero-knowledge protocol, a prover wants to convince a verifier that some statement is true, such that the verifier learns nothing except the validity of the assertion. Typically, the prover claims that an input string $u$ is in a language $L$, and after the interaction, the verifier accepts or rejects. We assume the reader is familiar with the basic theory of zero-knowledge protocols and just recall the most important notions informally: the protocol is an interactive zero-knowledge proof system for $L$ if it is complete, i.e., if $u \in L$, then the verifier accepts – and sound, i.e., if $u \notin L$ then no matter what the prover does, the verifier accepts with at most probability $\epsilon$, where $\epsilon$ is called the soundness error of the protocol. Finally, zero-knowledge means that given only that $u \in L$, conversations between the honest prover and an arbitrary poly-time verifier can be efficiently simulated and are indistinguishable from real conversations.

In this paper we concentrate on commitments to elements in a finite field $K$, or to integers and we assume that commitments are also homomorphic, i.e., both commitments and randomness are chosen from (finite) groups, and $[x] \cdot [y] = [x + y]$ (we will describe this property more in detail in section 2.1 and 7). For $K = \mathbb{F}_q$ for a prime $q$, such commitments can, for instance, be constructed from any $q$-invertible group homomorphism [CD98] that exists, if factoring or discrete log are hard problems. It is also easy based on known techniques – but perhaps less well known – that homomorphic commitments with unconditional hiding and binding can be built if we assume preprocessing, e.g., the committer gets random field elements and information theoretic MACs and the receiver gets corresponding keys. We give more details on this later (see Section 2.1). Finally, Homomorphic commitments to integers based on factoring were proposed in [FO97, DF02].

In typical applications of these commitment schemes, the prover needs to convince the verifier that the values he commits to satisfy a certain algebraic relation. A general way to state this is that the prover commits to $x_1, \ldots, x_v$, and the verifier wants to know that $D(x_1, \ldots, x_v) = 0$ for an algebraic circuit $D$ defined over $K$ or over the integers. If $D$ uses only linear operations, the verifier can himself compute a commitment to $D(x_1, \ldots, x_v)$ (using the homomorphic properties of the commitment scheme) and the prover opens this to reveal 0. However, if $D$ uses multiplication, we need a zero-knowledge protocol where the prover convinces the verifier that three committed values $x, y, z$ satisfy $xy = z$.

In [CDD+99], such a multiplication protocol was proposed for homomorphic commitments over any finite field $K$. The soundness error for that protocol is $1/|K|$, which is too large if $K$ is a field with small size (constant or logarithmic in the security parameter). The only known way to have a smaller error is to repeat the protocol. This solution leads to a protocol with communication complexity $\Theta(\kappa l)$ for soundness error $2^{-l}$ and where commitments have size $\kappa$ bits.

Likewise, a multiplication protocol for integer commitments was proposed in [FO97, DF02]. This protocol has essentially optimal communication complexity $\Theta(\kappa + l + k)$, where $k$ is size in bits of the prover’s secret integers, but it requires an extra assumption, namely the strong RSA assumption. If we only want to assume