Additive Combinatorics and Discrete Logarithm Based Range Protocols

***** Second e-print version*, October 3, 2009 *****

Rafik Chaabouni[1], Helger Lipmaa[2], and Abhi Shelat[3]

[1] EPFL, Switzerland
[2] Cybernetica AS, Estonia
[3] University of Virginia, USA

Abstract. We show how to express an arbitrary integer interval $\mathcal{I} = [0, H]$ as a sumset $\mathcal{I} = \sum_{i=1}^{\ell} G_i * [0, u-1] + [0, H']$ of smaller integer intervals for some small values ℓ, u, and $H' < u - 1$, where $b * A = \{ba : a \in A\}$ and $A + B = \{a + b : a \in A \wedge b \in B\}$. We show how to derive such expression of $\mathcal{I}$ as a sumset for any value of $1 < u < H$, and in particular, how the coefficients G_i can be found by using a non-trivial but efficient algorithm. This result may be interesting by itself in the context of additive combinatorics. Given the sumset-representation of $\mathcal{I}$, we show how to decrease the communication complexity of the recent range proof of Camenisch, Chaabouni and Shelat from ASIACRYPT 2008 by the factor of 2.

Keywords. Additive combinatorics, cryptographic range proof, sumset, zero knowledge.

1 Introduction

New Result in Additive Combinatorics. The principal contribution of this paper is to show that for any integer interval $\mathcal{I} = [0, H]$ and any $1 < u < H$, there is a sumset-representation

$$\mathcal{I} = \sum_{j=0}^{\ell-1} G_j * [0, u-1] + [0, H'] \tag{1}$$

for some $\ell \leq \lceil \log_u(H+1) \rceil$ and $H' \in [1, u-2]$, where $b * A = \{ba : a \in A\}$ and $A + B = \{a + b : a \in A \wedge b \in B\}$. We first derive a recursive formula for computing G_j for any $u > 1$. As an interesting technical contribution, we then show a semi-closed form for G_j, that is, we show how to compute G_j given only H, j and u. This algorithm is efficient and only requires simple arithmetic. More precisely, we show that G_j is equal to the sum of $\lfloor H/u^{j+1} \rfloor$ and a simple (but non-trivial) function of the $j+1$ lowest u-ary digits of H. We think that the presented algorithm may be interesting by itself say in the general context of additive combinatorics [TV06]: decompositions of sets as sumsets are common in additive combinatorics, but our concrete result differs significantly from existing results in that field.

* Corrected some small typos. The most annoying of them was that we had typos in the definition of $\boldsymbol{h}_{j+}$ and $[\![\boldsymbol{x}]\!]$ on page 6.

Range Proof: Background. We are now going to apply Eq. (1) in cryptography, more precisely in the construction of cryptographic range proofs. In a cryptographic range proof, the prover proves in zero knowledge that for given C and H, C is a commitment of some element $\sigma \in [0, H]$. (Modifying it to general ranges $[L, H]$ is trivial when one uses a homomorphic commitment scheme.) Range proofs are needed in various applications like e-voting [DJ01], e-auctions [LAN02], e-cash, etc. Range proofs with communication complexity $O(1)$ were introduced in [Bou00,Lip03]. However, such proofs work under very specific security assumptions, and thus there is still interest in protocols that are based on the discrete logarithm scenario. There exists a well-known folklore cryptographic range proof, see for example [DJ01], in the special case when $H = u^\ell - 1$ for some integers $u, \ell > 0$. In this protocol, the prover writes σ as $\sigma = \sum \sigma_j u^j$, commits—by using a homomorphic commitment scheme—to all values σ_j, and then proves in zero-knowledge (using say a Σ-protocol) that $\sigma_j \in [0, u-1]$ for all j. The folklore protocol is most efficient in the case $u = 2$, and thus this case has been studied the most. (For larger u, one has to use recursive range proofs for every σ_j.) Notice, however, that the communication complexity of this folklore range proof is $\Theta(\log H)$.

Recently, Camenisch, Chaabouni and Shelat [CCS08] presented a new range proof that works in the non-binary case. Assuming again $H = u^\ell - 1$, the verifier in their range proof first publishes signatures on all integers in $[0, u-1]$. The prover gives a proof of knowledge on signatures of ℓ committed elements σ_j. Analogously to the folklore protocol, this shows that the prover knows elements $\sigma_j \in [0, u-1]$ such that $\sigma = \sum \sigma_j u^j$ (the latter part is trivial with a public homomorphic commitment to σ).

However, if $H \neq u^\ell - 1$ then both the folklore protocol and the protocol of [CCS08] get more complicated, and require up to 2 times more communication. (In a nutshell, this is because they show that $\sigma \in [0, H]$ by using an AND composition of two range proofs of form $\sigma' \in [0, u^\ell - 1]$.) In the special case $u = 2$, an efficient modification of the folklore case for general ranges was proposed (though its correctness was not proven) in [LAN02]. There it was noted that for any $H \geq 1$, $\sigma \in [0, H]$ if and only if $\sigma = \sum_{j=0}^{\lfloor \log_2 H \rfloor} G_j \sigma_j$, where $\sigma_j \in \{0, 1\}$ and $G_j := \left\lfloor (H + 2^j)/2^{j+1} \right\rfloor$. For example, $\sigma \in [0, 11]$ iff $\sigma = 6\sigma_0 + 3\sigma_1 + \sigma_2 + \sigma_3$ for $\sigma_j \in \{0, 1\}$. Thus the folklore protocol can be extended to arbitrary values of H with virtually no efficiency loss. In the language of additive combinatorics, the result of [LAN02] says that

$$[0, H] = \sum \left\lfloor (H + 2^j)/2^{j+1} \right\rfloor * [0, 1] \ . \tag{2}$$

Note that Eq. (2) does not straightforwardly generalize to the case where we are interested in a larger range $[0, u-1]$. In fact, [LAN02] did not even present a proof that Eq. (2) holds. As a straightforward corollary of our sumset-representation of $[0, H]$, we obtain a proof that the presentation of Eq. (2) is correct.

Application of the Sumset-Representation in Range Proofs. We show how to use the sumset-representation Eq. (1) to modify the range proof of [CCS08] so that it will become at least 50% more communication-efficient in practice (and so that it is *always* more efficient than the folklore protocol). For this we use a simple corollary of our general sumset-representation that $[0, H] = \sum_{j=0}^{\ell-1} G_j * [0, u-1]$ whenever $(u-1) | H$.

Moreover, if we set $u = O(\log H/\log\log H)$, then the total communication of the range proof is $\Theta(\log H/\log\log H)$. We also point out some mistakes in [CCS08], namely, that the so called OR composition proposed there does not work in most of the cases, and thus their protocols are somewhat less efficient than claimed.

Finally, our secondary contribution is an implicit framework for generating range proofs. Recall that when the group order is hidden (e.g., in RSA-like settings), earlier range protocols [Lip03] exploit the quadratic form

$$y = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

to represent any positive integer. Since there is a 4-term quadratic form, range proofs in this setting have constant size. As discussed below, this idea does not help when the group order is known (e.g. under discrete-log like assumptions). Nonetheless, finding more efficient linear or quadratic forms for ranges $[0, H]$, in which each term is in a small range will immediately lead to more efficient range proofs in the discrete logarithm setting.

2 Preliminaries

We summarize and copy some of the notation and definitions from [CCS08] for consistency and to make it easier for the reader to follow.

Notation. PPT means probabilistic polynomial-time. k is the security parameter. In all protocols, prover and verifier send elements from $\mathbb{G}_1$, $\mathbb{G}_T$ and $\mathbb{Z}_p$. We denote the length of representation (which may differ from the logarithm of the cardinality of the groups) of such elements by $\mathsf{replen}(\mathbb{G}_1)$, $\mathsf{replen}(\mathbb{G}_T)$ and $\mathsf{replen}(\mathbb{Z}_p)$ respectively. For any two integers $L \leq H$, let $[L, H] := \{x \in \mathbb{Z} : L \leq x \leq H\}$. We use the usual "set-theoretic" arithmetic notation. For example, if A and B are sets then $A + B = \{a + b : a \in A \wedge b \in B\}$. Moreover, for an integer b and $A \subset \mathbb{Z}$, $b * A = \{ba : a \in A\}$, this is also called the b-dilate of A [TV06].

Commitment Schemes. A (string) commitment scheme is a triple of algorithms $C = (\mathrm{Gen}, \mathrm{Com}, \mathrm{Open})$ representing the generation, the commit and the open algorithm. The Gen algorithm generates parameters p for a scheme. The Com algorithm runs on input (p, m, r) where m is a string, and r is a random tape, and produces a pair of values (c, o) representing the committed string and an opening string. The Open algorithm runs on input (c, m, o) and outputs 0 or 1. The scheme should have a "hiding" property and a "binding" property which informally require it to be difficult (or impossible) for the adversary to determine the message m from c or to open the value of a commitment c to two different messages m_1, m_2.

Zero-Knowledge Proofs and Σ-Protocols. We use definitions from [BG92,CDM00]. A pair of interacting algorithms (P, V) is a proof of knowledge (PK) for a relation $R = \{(\alpha, \beta)\} \subseteq \{0,1\}^* \times \{0,1\}^*$ with knowledge error $\kappa \in [0, 1]$ if (1) for all

$(\alpha, \beta) \in R$, $\mathsf{V}(\alpha)$ accepts a conversation with $\mathsf{P}(\beta)$ with probability 1; and (2) there exists an expected polynomial-time algorithm E, called the *knowledge extractor*, such that if a cheating prover P^* has probability ϵ of convincing V to accept α, then E, when given rewindable black-box access to P^*, outputs a witness β for α with probability $\epsilon - \kappa$.

A proof system (P, V) is *computational honest-verifier zero-knowledge* if there exists a PPT algorithm Sim, called the *simulator*, such that for any $(\alpha, \beta) \in R$, the outputs of $V(\alpha)$ after interacting with $\mathsf{P}(\beta)$ and that of $\mathsf{Sim}(\alpha)$ are computationally indistinguishable. When we will talk about *honest-verifier zero-knowledge* we will assume the computational case.

Note that standard techniques can be used to transform an honest-verifier zero-knowledge proof system into a general zero-knowledge one [CDM00]. This is especially true of special Σ-protocols that will be presented later in the paper. Thus, for the remainder of the paper, our proofs will be honest-verifier zero-knowledge. (This also allows us to make more accurate comparisons with the other proof techniques since they are usually also presented as honest-verifier protocols.)

A Σ-protocol is a proof system (P, V) where the conversation is of the form (a, c, z), where a and z are computed by P, and c is a challenge randomly chosen by V. The verifier accepts if $\phi(\alpha, a, c, z) = 1$ for some efficiently computable predicate ϕ. A Σ-protocol must satisfy three security requirements: correctness, special soundness and special honest-verifier zero knowledge (SHVZK). A Σ-protocol is correct when a honest prover convinces honest verifier with probability $1 - k^{-\omega(1)}$. A Σ-protocol has the special soundness property when from two accepting views (a, c, z) and (a, c', z'), where $c \neq c'$, one can efficiently recover a witness w such that $w \Rightarrow x \in L$. A Σ-protocol has the SHVZK property if there exists a PPT simulator Sim that can first randomly pick c^*, z^* (from some fixed sets) and then compute an a^* such that the view (a^*, c^*, z^*) is accepting and the distribution (a^*, c^*, z^*) is computationally indistinguishable from the distribution of accepting views between honest server and honest verifier.

We use the notation introduced by Camenisch and Stadler [CS97] for various zero-knowledge proofs of knowledge of discrete logarithms and proofs of the validity of statements about discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \gamma) : y = g^{\alpha} h^{\beta} \ \wedge \ \mathfrak{y} = \mathfrak{g}^{\alpha} \mathfrak{h}^{\gamma} \ \wedge \ (u \leq \alpha \leq v)\}$$

denotes a "*zero-knowledge* P*roof of* K*nowledge of integers* α*,* β*, and* γ *such that* $y = g^{\alpha} h^{\beta}$ *and* $\mathfrak{y} = \mathfrak{g}^{\alpha} \mathfrak{h}^{\gamma}$ *holds, where* $v \leq \alpha \leq u$," where $y, g, h, \mathfrak{y}, \mathfrak{g}$, and $\mathfrak{h}$ are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\mathfrak{G} = \langle \mathfrak{g} \rangle = \langle \mathfrak{h} \rangle$. The convention is that Greek letters denote quantities the knowledge of which is being proved, while all other parameters are known to the verifier. Using this notation, a proof-protocol can be described by just pointing out its aim while hiding all details. We note that all of the protocols we present in this notation can be easily instantiated as Σ-protocols.

Definition 1 (Proof of Set Membership [CCS08]).
Let $C = (\mathrm{Gen}, \mathrm{Com}, \mathrm{Open})$ *be the generation, the commit and the open algorithm of a string commitment scheme. For an instance* c*, a proof of set membership with respect*

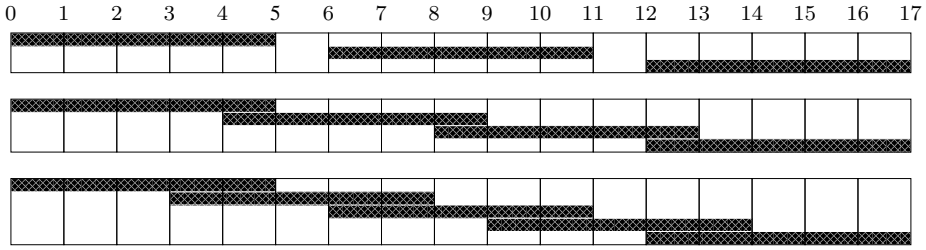

Fig. 1. Illustration of the first recursive step of Thm. 1. Here $H = H_0 = 17$, and $u \in \{3, 4, 5\}$. For example, in the top graph, $u = 3$, $G_0 = \lfloor (17+1)/3 \rfloor = 6$, and $H_1 = 17 - 2 \cdot 6 = 5$

to commitment scheme C and set Φ is a proof of knowledge for the following statement:

$$PK\{(\sigma, \rho) \ : \ c \leftarrow \mathrm{Com}(\sigma;\ \rho) \wedge \sigma \in \Phi\}$$

Definition 2 (Range Proof [CCS08]). *A range proof with respect to a commitment scheme C is a special case of a proof of set membership in which the set Φ is a continuous sequence of integers $\Phi = [a, b]$ for $a, b \in \mathbb{N}$.*

As discussed in the introduction, some efficient range proofs were proposed in [Bou00,LAN02,Lip03,CCS08]. We will give a precise description of the proof from [CCS08] in Sect. 4.

3 Sumset-Representation of Integer Intervals

The goal of this section is to derive a sumset-representation $[0, H] = \sum_{i=0}^{\ell-1} G_i * [0, u-1] + [0, H']$, where $1 < H' < u \ll H$, of an arbitrary integral interval $[0, H]$. (Integral means that all involved parameters H, H', u and G_i are positive integers.) Moreover, we aim to find minimal ℓ for any fixed value of u.

First we give an intuitive derivation of our result. (See also Fig. 1.) Fix H and u. Let $H_0 = H$. Then clearly $[0, H_0] = G_0 * [0, u-1] + [0, H_1]$, where $G_0 := \lfloor (H_0+1)/u \rfloor$ and $H_1 = H_0 - (u-1) \cdot G_0$. This can be derived as follows: we want to divide $[0, H_0]$ into u smaller (possibly overlapping) intervals of equal size H_1 such that H_1 is minimal. The sub-intervals should start at periodic positions jG_0, for some G_0. Because all elements from $[0, H_0]$ must belong to at least one of those subareas, it must be the case that $H_1 \geq G_0 - 1$ and $(u-1)G_0 + H_1 = H_0$. Thus, in the optimal case when $H_1 = G_0 - 1$, we get $uG_0 - 1 = H_0$ or $G_0 = (H_0+1)/u$. Since G_0 has to be an integer, we set $G_0 = \lfloor (H_0+1)/u \rfloor$. Finally, $H_1 = H_0 - (u-1)G_0$ as stated.
These formulas reduce the case $[0, H_0]$ to a smaller case $[0, H_1]$ that can be solved similarly. Recursively,

$$[0, H] = [0, u-1] \cdot \sum_j G_j + [0, H'] \ ,$$

where

$$G_j := \left\lfloor \frac{H_j + 1}{u} \right\rfloor \quad . \tag{3}$$

and

$$H_{j+1} := H_j - (u-1) \cdot G_j = H_j - (u-1) \cdot \left\lfloor \frac{H_j + 1}{u} \right\rfloor \quad .$$

This process stops when the interval $[0, H_{j+1}]$ is small enough so that it cannot be covered by u different non-empty intervals, that is, if $H_{j+1} \leq u - 1$. Then we define $\ell(u, H) := j + 1$ to be the number of steps in this recursive process. Clearly, after we are done with the recursive process,

$$H' := H_{j+1} = H - \left\lfloor \frac{H}{u-1} \right\rfloor \cdot (u-1) \quad .$$

This means in particular that if $(u-1) \mid H$ then $H' = 0$.

Example 1. For example, with $H = 57$ and $u = 4$, one can verify that $[0, 57] = 14 * [0, 3] + 4 * [0, 3] + [0, 3]$. As another example, $[0, 160] = 40 * [0, 3] + [0, 40] = 40 * [0, 3] + 10 * [0, 3] + [0, 10] = 40 * [0, 3] + 10 * [0, 3] + 2 * [0, 3] + [0, 4] = 40*[0, 3]+10*[0, 3]+2*[0, 3]+1*[0, 3]+[0, 1]$. Now we are done since $1 < u-1 = 3$.

(Another, though non-recursive, example with $H = 17$ was already depicted by Fig. 1.)

Finally, the sequence $(\dots, G_j, \dots)$ clearly decreases the slowest when for all j, $u \mid (H_{j+1} + 1)$, since then the floor operation is not applied. But $u \mid (H_{j+1} + 1)$ iff $u \mid (H_j - (u-1)G_j + 1)$ iff $u \mid (H_j + G_j + 1)$ iff (because also $u \mid (H_j + 1)$) $u \mid G_j$. Thus, the sequence is slowest to decrease if $H + 1 = u^\ell$ for some ℓ. This means, that the process is guaranteed to stop in $\ell(u, H) \leq \log_u(H + 1)$ steps.

This leads us to the following theorem.

Theorem 1. *Let $u \geq 2$, $H \geq u$. Let G_j, H_j and H' be defined as before. Denote $\ell = \ell(u, H) \leq \lceil \log_u(H+1) \rceil$ as above. Then $[0, H] = \sum_{j=0}^{\ell-1} G_j * [0, u-1] + [0, H']$. If $(u-1) \mid H$ then $H' = 0$.*

Proof. Clear from above. □

Semi-Closed Form for G_j. While the presented recursive formulas for G_j and H_{j+1} are efficient, it is desirable to to have a closed form for G_j. In the next we show a semi-closed form, that is, a formula for G_j that only depends on u, j and H.

Assume that $H = \sum h_j u^j$ with $h_j \in \{0, \dots, u-1\}$. For any j, write $h_{j+} := \lfloor H/u^j \rfloor$, that is, $H = u^j h_{j+} + \sum_{i=0}^{j-1} u^i h_i$. In particular, $h_{j+} = u h_{(j+1)+} + h_j$. Define $[\![x]\!] := x \pmod{u-1}$. Our proof is built up on the initial observation that

$$G_0 = h_{1+} + \left\lfloor \frac{h_0 + 1}{u} \right\rfloor \quad ,$$

$$H_1 = h_{1+} + h_0 - (u-1) \left\lfloor \frac{1 + h_0}{u} \right\rfloor = h_{1+} + [\![h_0]\!] \quad .$$

The latter equation is obvious: if $h_0 < u-1$ then $h_0 - (u-1)\left\lfloor \frac{1+h_0}{u} \right\rfloor = h_0 = [\![h_0]\!]$ and if $h_0 = u-1$ then $h_0 - (u-1)\left\lfloor \frac{1+h_0}{u} \right\rfloor = u - 1 - (u-1) = 0 = [\![h_0]\!]$. We can now prove that

Theorem 2. $G_j = h_{(j+1)+} + \left\lfloor \frac{h_j + [\![\sum_{i=0}^{j-1} h_i]\!] + 1}{u} \right\rfloor$.

Proof. By induction. We prove that $H_j = h_{(j+1)+} + \left[\!\!\left[\sum_{i=0}^{j} h_i\right]\!\!\right]$, from this the claim for G_j is obvious. Induction basis ($j = 0$) is obvious since $H_0 = h_{0+}$.

Induction step ($j > 0$). Assume that $H_j = h_{j+} + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] = uh_{(j+1)+} + h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right]$ and $G_j = h_{(j+1)+} + \left\lfloor \frac{h_j + [\![\sum_{i=0}^{j-1} h_i]\!] + 1}{u} \right\rfloor$. Then

$$H_{j+1} = H_j - (u-1)G_j = h_{(j+1)+} + h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] - (u-1)\cdot \left\lfloor \frac{h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + 1}{u} \right\rfloor .$$

Thus to finish the proof we only have to show that

$$h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] - (u-1)\cdot \left\lfloor \frac{h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + 1}{u} \right\rfloor = \left[\!\!\left[\sum_{i=0}^{j} h_i\right]\!\!\right] \tag{4}$$

for any $h_i \in \{0, \ldots, u-1\}$. We consider the next cases.

Case 1, $\left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] = 0$. Then the left hand side of Eq. (4) is $h_j - (u-1)\cdot \lfloor (1+h_j)/u \rfloor = [\![h_j]\!]$ and the right hand side is equal to the same value.

Case 1, $\left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] \neq 0$ and $h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + 1 < u$. Then the left hand side of Eq. (4) is $h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right]$ and the right hand side is $\left[\!\!\left[\sum_{i=0}^{j-1} h_i + h_j\right]\!\!\right] = \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + h_j$.

Case 3, $\left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] \neq 0$ and $h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + 1 \geq u$. Then the left hand side of Eq. (4) is $h_j + \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] - (u-1)$ and the right hand side is $\left[\!\!\left[\sum_{i=0}^{j-1} h_i + h_j\right]\!\!\right] = \left[\!\!\left[\sum_{i=0}^{j-1} h_i\right]\!\!\right] + h_j - (u-1)$. □

In the binary case $u = 2$, a formula like this was already given in [LAN02]. However, while [LAN02] stated the closed form, they did not prove it. Fortunately, their formula follows straightforwardly from the general result.

Corollary 1 (Binary case, [LAN02]). *If* $u = 2$ *then* $G_j = h_{(j+1)+} + \left\lfloor \frac{h_j+1}{u} \right\rfloor = \left\lfloor \frac{H+2^j}{2^{j+1}} \right\rfloor$.

Proof. Straightforward corollary.

4 Preliminaries: CCS Range Proof

Computational Assumptions. The following protocols require bilinear groups and associated hardness assumptions. These assumptions are summarized from [CCS08].

Let PG be a pairing group generator that on input 1^k outputs descriptions of multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_\mathrm{T}$ of prime order p where $|p| = k$. Let $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{1\}$ and let $g \in \mathbb{G}_1^*$. The generated groups are such that there exists an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathrm{T}$, meaning that (1) for all $a, b \in \mathbb{Z}_p$ it holds that $e(g^a, g^b) = e(g, g)^{ab}$; (2) $e(g, g) \neq 1$; and (3) the bilinear map is efficiently computable.

Definition 3 (Strong Diffie-Hellman Assumption [BB04]). *We say that the q-SDH assumption associated to a pairing generator PG holds if for all PPT adversaries A, the probability that $A(g, g^x, \ldots, g^{x^q})$ where $(\mathbb{G}_1, \mathbb{G}_\mathrm{T}) \leftarrow \mathsf{PG}(1^k)$, $g \leftarrow \mathbb{G}_1^*$ and $x \leftarrow \mathbb{Z}_p$, outputs a pair $(c, g^{1/(x+c)})$ where $c \in \mathbb{Z}_p$ is negligible in k.*

As noted by [CCS08], Cheon's [Che06] attack against this type of assumption is not relevant if $q \leq 50$ as it is in this protocol.

Boneh-Boyen Signatures. Our scheme relies on the elegant Boneh-Boyen short signature scheme [BB04] which we briefly summarize. The signer's secret key is $x \leftarrow \mathbb{Z}_p$, the corresponding public key is $y = g^x$. The signature on a message m is $\sigma \leftarrow g^{1/(x+m)}$; verification is done by checking that $e(\sigma, y \cdot g^m) = e(g, g)$. This scheme is similar to the Dodis and Yampolskiy verifiable random function [DY05].

Security under weak chosen-message attack is defined through the following game. The adversary begins by outputting ℓ messages $m_1, \ldots, m_\ell$. The challenger generates a fresh key pair and gives the public key to the adversary, together with signatures $\sigma_1, \ldots, \sigma_\ell$ on $m_1, \ldots, m_\ell$. The adversary wins if it succeeds in outputting a valid signature σ on a message $m \notin \{m_1, \ldots, m_\ell\}$. The scheme is said to be unforgeable under a chosen-message attack if no p.p.t. adversary A has non-negligible probability of winning this game. Our scheme relies on the following property of the Boneh-Boyen short signature [BB04] which we paraphrase below:

Lemma 1 ([BB04]). *Suppose the q-Strong Diffie Hellman assumption holds in $(\mathbb{G}_1, \mathbb{G}_\mathrm{T})$. Then the basic Boneh-Boyen signature scheme is q-secure against an existential forgery under a weak chosen message attack.*

The Camenisch-Chaabouni-Shelat range proof in the case when $H = u^\ell - 1$ is depicted by Prot. 1. In particular, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is an admissible bilinear map for some multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$, and g is a generator of $\mathbb{G}_1$ with $h \in \langle g \rangle$.

Communication of CCS Range Proof for "Nice" H. The CCS range proof for nice H requires the prover to compute 3ℓ exponentiations and 2ℓ pairings (in [CCS08], this was summed up as 5ℓ exponentiations). It requires non-interactive (static) communication of

$$\mathsf{NICom}_{\mathsf{ccs}}(u, \ell) := (1 + u) \cdot \mathsf{replen}(\mathbb{G}_1) \ ,$$

Assume $\sigma = \sum_{j=0}^{\lfloor \log_u (H+1) \rfloor} \sigma_j u^j$.
Common input: g, h, u, ℓ, and a commitment C.
Prover's input: σ, r such that $C = g^\sigma h^r$ and $\sigma \in [0, H]$.

1. **The verifier does:** generate a random $x \leftarrow \mathbb{Z}_p$, and set $y \leftarrow g^x$. For $i \in [0, u-1]$, set $A_i \leftarrow g^{1/(x+i)} \in \mathbb{G}_1$. She sends $(y, A_0, \ldots, A_{u-1})$ to the prover.
2. **The prover does:** For all $j \in [0, \ell-1]$, generate random $v_j \leftarrow \mathbb{Z}_p$, set $V_j \leftarrow A_{\sigma_j}^{v_j} \in \mathbb{G}_1$. He sends $(V_0, \ldots, V_{\ell-1})$ to the verifier.
3. **The prover** uses the next Σ-protocol to prove to the verifier that $C = h^r \cdot g^{\sum \sigma_j u^j}$, and $V_j = g^{v_j/(x+\sigma_j)}$ for all j:
   (a) **The prover** picks $s_j, t_j, m_j \leftarrow \mathbb{Z}_p$ for $j \in [0, \ell-1]$. He sets $a_j \leftarrow e(V_j, g)^{-s_j} e(g, g)^{t_j} \in \mathbb{G}_T$, for $j \in [0, \ell-1]$, and $D \leftarrow g^{\sum_j u^j s_j} \cdot h^{\sum_j m_j} \in \mathbb{G}_1$. He sends $(a_0, \ldots, a_{\ell-1}, D)$ to the verifier.
   (b) **The verifier** sends a random challenge $c \leftarrow \mathbb{Z}_p$ to the prover.
   (c) **The prover** sets $z_j^{(\sigma)} \leftarrow s_j - \sigma_j c \mod p$, $z_j^{(v)} \leftarrow t_j - v_j c \mod p$, for $j \in [0, \ell-1]$. He sets $z^{(m)} \leftarrow m - rc \mod p$, where $m = \sum_{j=0}^{\ell-1} m_j$. He sends $(z_0^{(\sigma)}, \ldots, z_{\ell-1}^{(\sigma)}, z_0^{(v)}, \ldots, z_{\ell-1}^{(v)}, z^{(m)})$ to the verifier.
   (d) **The verifier** checks that $D = C^c h^{z^{(m)}} g^{\sum_j u^j \cdot z_j^{(\sigma)}}$ and $a_j = e(V_j, y)^c \cdot e(V_j, g)^{-z_j^{(\sigma)}} \cdot e(g, g)^{z_j^{(v)}}$ for every $j \in [0, \ell-1]$.

Protocol 1: The CCS cryptographic range proof for range $[0, u^\ell - 1]$.

bits (signatures and public keys that can be shared between different protocol runs), and interactive communication (which is unique for every protocol run) of

$$\mathsf{ICom}_{\mathsf{ccs}}(u, \ell) := (1 + \ell) \cdot \mathsf{replen}(\mathbb{G}_1) + \ell \cdot \mathsf{replen}(\mathbb{G}_T) + (2 + 2\ell) \cdot \mathsf{replen}(\mathbb{Z}_p)$$

bits.

Communication of CCS for Arbitrary Range $[L, H]$. As noted in [CCS08], to prove that $\sigma \in [L, H]$ for arbitrary L and H, one can use an AND composition. More precisely, suppose that $u^{\ell-1} < H < u^\ell$. Then to show that $\sigma \in [L, H]$, it suffices to show that $\sigma \in [L, L + u^\ell)$ and $\sigma \in [H - u^\ell, H)$. Equivalently, one has to show that $\sigma - L \in [0, u^\ell)$ and $\sigma - H + u^\ell \in [0, u^\ell)$.

For this, one uses the standard AND composition of Prot. 1 with itself. Recall that an AND composition of two Σ-protocols A_1 and A_2 is a Σ protocol where the first message is a composition of the first messages of A_1 and A_2, the second message is a single challenge c, and the third message is a composition of the third messages of A_1 and A_2 that correspond to the first messages and the single challenge c. Moreover, static information (the public key y and all signatures) and also the values V_j are only sent once. Thus, in the AND composition of the CCS protocol, there are two versions of a_j, D, $z_j^{(\sigma)}$, $z_j^{(v)}$ and $z^{(m)}$, which makes the (static) communication of the AND composition of Prot. 1 with itself equal to

$$\mathsf{NICom}_{\mathsf{ccsand}}(u, \ell) = \mathsf{NICom}_{\mathsf{ccs}}(u, \ell) = (1 + u) \cdot \mathsf{replen}(\mathbb{G}_1) \ ,$$

and the dynamic communication is equal to

$$\begin{aligned}\mathsf{ICom}_{\mathsf{ccsand}}(u,\ell) =& \mathsf{ICom}_{\mathsf{ccs}}(u,\ell) + \ell \cdot \mathsf{replen}(\mathbb{G}_T) + \mathsf{replen}(\mathbb{G}_1) + (2\ell+1)\cdot \mathsf{replen}(\mathbb{Z}_p)\\ =& (\ell+2)\cdot \mathsf{replen}(\mathbb{G}_1) + 2\ell \cdot \mathsf{replen}(\mathbb{G}_T) + (4\ell+3)\cdot \mathsf{replen}(\mathbb{Z}_p)\ .\end{aligned}$$

Remark on OR Composition. [CCS08] also considered the OR composition. The communication of an OR composition is twice the communication of the single protocol, but with $\ell - 1$ instead of ℓ, and thus the OR composition has the potential to be more efficient than the AND composition. In our case, for the OR composition to work, we have to assume that u is such that $u^{\ell-1} < H \leq 2 \cdot u^{\ell-1}$. In this case, $\sigma \in [0, H]$ iff $\sigma \in [0, u^{\ell-1}]$ or $\sigma \in [H - u^{\ell-1}, H]$. This differs slightly from the misstated requirement of [CCS08], where it was said that one just needs that $u^{\ell-1} < H$. In particular, this means that the OR composition does not work for values, considered in Sect. 4.4 of [CCS08], and thus the communication-efficiency of their range proof is (in most of the cases) slightly worse than claimed in [CCS08].

Communication Analysis. Let us assume that $\mathsf{replen}(\mathbb{G}_T) \approx 12 * \mathsf{replen}(\mathbb{G}_1) \approx \mathsf{replen}(\mathbb{Z}_p)$ [GPS08]. Following [CCS08] and plugging in parameters in terms of the $\mathsf{replen}(\mathbb{G}_1)$, the communication can then be minimized by solving the following system

$$\min\ 6 + u + 29\ell \ \text{ s.t. } u^{\ell} \geqslant H$$

Setting $u = \frac{\log H}{\log\log H}$ then we get a total asymptotic communication complexity of

$$Com(u,\ell) = O\left(\frac{\log H}{\log\log H - \log\log\log H}\right)$$

which is asymptotically smaller than $O(\log H)$. For concrete parameters, we substitute the constraint that $u^{\ell} \approx H$ into the equation $u + \ell$ above, set the derivative with respect to u to 0 and attempt to solve the equation:

$$1 - \frac{29\log H}{u\log^2 u} = 0$$

which simplifies to

$$u\log^2 u = 29\log H. \tag{5}$$

This equation cannot be solved analytically. However, given H, we can use numerical methods to find a good u as described in [Bla97].

5 Modified Range Proof: New

The idea of the next proof follows from Thm. 1. We can assume that $u > 1$. Clearly, $\sigma \in [0, H]$ iff

$$(u-1)\sigma \in [0, (u-1)H]$$

Assume $(u-1)\cdot\sigma = \sum_{j=0}^{\ell-1} \sigma_j \cdot G_j$ for $\ell = \ell(u,(u-1)H) \leq \lceil \log_u((u-1)\cdot H+1)\rceil$.
Common input: g, h, u, ℓ, and a commitment C.
Prover's input: σ, r such that $C = g^{\sigma}h^{r}$ and $\sigma \in [0, H]$.

1. **The verifier does:** she generates a random $x \leftarrow \mathbb{Z}_p$, and sets $y \leftarrow g^x$. For $i \in [0, u-1]$, she sets $A_i \leftarrow g^{1/(x+i)} \in \mathbb{G}_1$. She sends $(y, A_0, \ldots, A_{u-1})$ to the prover.
2. **The prover does:** For all $j \in [0, \ell-1]$, generate random $v_j \leftarrow \mathbb{Z}_p$, set $V_j \leftarrow A_{\sigma_j}^{v_j} \in \mathbb{G}_1$. He sends $(V_1, \ldots, V_{\ell-1})$ to the verifier.
3. **The prover** uses the next Σ-protocol to prove to the verifier that $C^{u-1} = h^{(u-1)\cdot r} \cdot g^{\sum \sigma_j G_j}$, and $V_j = g^{v_j/(x+\sigma_j)}$ for all $j \in [0, \ell-1]$:
   (a) **The prover** picks $s_j, t_j, m_j \leftarrow \mathbb{Z}_p$ for $j \in [0, \ell-1]$. He sets $a_j \leftarrow e(V_j, g)^{-s_j} e(g,g)^{t_j} \in \mathbb{G}_T$, for $j \in [0, \ell-1]$, and $D \leftarrow g^{\sum_j s_j \cdot G_j} \cdot h^{(u-1)\cdot \sum_j m_j} \in \mathbb{G}_1$. He sends $(a_0, \ldots, a_{\ell-1}, D)$ to the verifier.
   (b) **The verifier** sends a random challenge $c \leftarrow \mathbb{Z}_p$ to the prover.
   (c) **The prover** sets $z_j^{(\sigma)} \leftarrow s_j - \sigma_j c$, $z_j^{(v)} \leftarrow t_j - v_j c$ for $j \in [0, \ell-1]$ and $z^{(m)} \leftarrow m - rc$ for $m = \sum_j m_j$. He sends $(z_0^{(\sigma)}, \ldots, z_{\ell-1}^{(\sigma)}, z_0^{(v)}, z_{\ell-1}^{(v)}, z^{(m)})$ to the verifier.
   (d) **The verifier** checks that $D = C^{c(u-1)} \cdot h^{(u-1)\cdot z^{(m)}} \cdot g^{\sum_j z_j^{(\sigma)} \cdot G_j}$ and $a_j = e(V_j, y)^c \cdot e(V_j, g)^{-z_j^{(\sigma)}} \cdot e(g,g)^{z_j^{(v)}}$ for every $j \in [0, \ell-1]$.

Protocol 2: New, generalization of CCS protocol for arbitrary range $[0, H]$.

iff, because of Thm. 1,

$$(u-1)\sigma = \sum_{j=0}^{\ell(u,(u-1)H)-1} \sigma_j G_j$$

for some $\sigma_j \in [0, u-1]$, and G_j are defined as in Thm. 1 with $H_0 = (u-1)H$.

Thus, we can propose a new range proof where we prove that C^{u-1} commits a value in $(0, (u-1)H]$ by using the CCS protocol for "nice" H, see Prot. 2. Note that changing 0 to any meaningful L, $1 \leq L < (u-1)H$, is trivial. In the description of the protocol, see Prot. 2, new parts (compared to the CCS protocol) have been colored red for easy parsing.

Rationale for multiplying by $u-1$. If $(u-1)$ divides H, then it is not necessary to multiply by the commitment by $(u-1)$. Recall that if $u-1$ does not divide H, then $H' < u$ (the leftover value) defines some small range $[0, H']$. In this case, one could (instead of multiplying by $u-1$) add an extra step to the range proof that shows that some new committed element belongs to the range $[0, H']$. Doing this would require an extra $H'+1$ elements from $\mathsf{replen}(\mathbb{G}_1)$, one extra element from $\mathsf{replen}(\mathbb{G}_T)$, and one extra element from $\mathsf{replen}(\mathbb{Z}_p)$ to be transmitted. Thus, it will always be more expensive to add this extra step. Another idea might be to use a simple OR-proof to handle the last $[0, H']$ elements. This would require extra communication of $H' \cdot \mathsf{replen}(\mathbb{G}_1) + (1 + H') \cdot \mathsf{replen}(\mathbb{Z}_p)$ bits. Since one element of $\mathsf{replen}(\mathbb{G}_T)$ is roughly 12 times larger than the size of one element from either $\mathbb{G}_1$ or $\mathbb{Z}_p$, this approach is favorable when $H' < 6$.

Theorem 3. *Assuming the q-SDH assumption, Prot. 2 is correct and has the property of special soundness and SHVZK.*

Proof (Sketch.). The proof is a straightforward extension of the security proof from [CCS08]. □

Concrete Efficiency of New. Clearly, both the static and dynamic communication of New is related to communication of the CCS protocol in the next simple way:

$$\mathsf{NICom}_{\mathsf{New}}(u,\ell) := \mathsf{NICom}_{\mathsf{ccs}}(u,\ell(u,(u-1)(H+1)))$$

and

$$\mathsf{ICom}_{\mathsf{New}}(u,\ell) := \mathsf{ICom}_{\mathsf{ccs}}(u,\ell(u,(u-1)(H+1)))\ .$$

This is easily seen to be a factor of 2 more efficient than having to use two proofs to handle an arbitrary range H.

Efficiency of New. Asymptotically, the *total* communication $\mathsf{NICom}_{\mathsf{New}} + \mathsf{ICom}_{\mathsf{New}}$ of remains the same:

$$\begin{aligned} u + \ell(u,(u-1)(H+1)) &\leq u + \log_u((u-1)(H+1)) \\ &= u + \log_u(u-1) + \log_u(H+1) \\ &\leq u + 1 + \log_u(H) + \frac{1}{H} \end{aligned}$$

As before, this value is (approximately) minimized when we set $u \leftarrow \frac{\log_2 H}{\log_2 \log_2 H}$. Concretely, there is a factor of two difference. The communication can be minimized by solving

$$\min 4 + u + 15\ell \quad \text{such that} \quad \ell > \log_u((u-1)(H+1))$$

As mentioned before, in some cases when $u-1$ already divides H, it is not necessary to multiply by $u-1$; even when $(u-1)$ does not evenly divide H, a standard OR-proof can sometimes be used to handle H'. We take this fact in account when computing the protocol's efficiency for a given range below. In the graph below, we show how the complexity of our new protocol compares with that of [CCS08] for ranges $[0,H]$ where H varies from 1000 to $2 \cdot 10^8$.

5.1 Comparison of Case Analysis

As a second way to compare the new protocol with protocol from [CCS08] and other previous work, we use the same numbers as in Sect. 4 of [CCS08]. In particular we assume that the size of $\mathbb{G}_1$ is 256 bits, the size of $\mathbb{G}_T$ is 3072 bits and the size of $\mathbb{Z}_p$ is upper-bounded by 256 bits. We also use the range $\mathcal{R} = [L,H) = [347184000, 599644800)$ as in [CCS08]. Also, clearly, the new protocol (as is the CCS protocol) for $\mathcal{R}$ is exactly as efficient as protocol for range $[0,H']$, where $H' = H - L - 1$. That is, $H' = 252460799$.

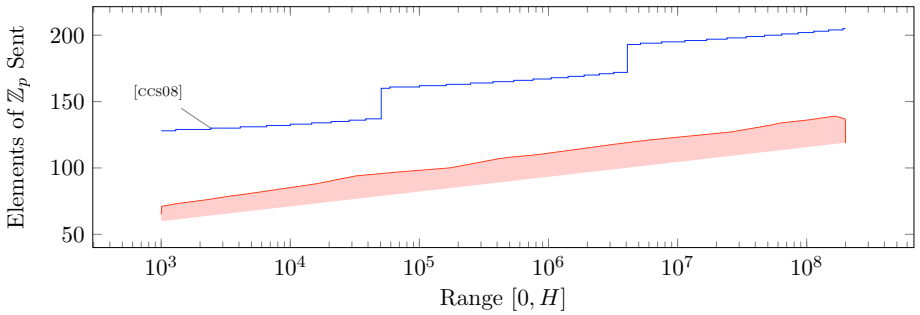

Fig. 2. Relative Efficiency of the New Protocol vs. [CCS08]. The number of group elements are computed under the assumption that $\mathsf{replen}(\mathbb{G}_T) \approx 12 * \mathsf{replen}(\mathbb{G}_1) \approx \mathsf{replen}(\mathbb{Z}_p)$. The complexity of our new protocol depends more sensitively on the exact value of H; therefore the red shaded area represents the convex hull of the values for our new protocol. The vertical gaps in the curve for [CCS08] are a result of the ratio 12 used above.

The values of $\mathsf{NICom}_{\mathsf{ccsand}}$, $\mathsf{ICom}_{\mathsf{ccsand}}$, $\mathsf{NICom}_{\mathsf{New}}$ and $\mathsf{NICom}_{\mathsf{New}}$ for a few different choices of u and ℓ are shown in the next two tables. Note that the optimal choice of u depends on how many times the range proof is going to be reused: the larger is the number w of reuses, the larger should be u, and for w reuses, one should choose a value of u for which $\mathsf{NICom}_{\mathsf{New}}(u, \ell) + (w-1)\mathsf{ICom}_{\mathsf{New}}(u, \ell)$ is minimal.

The values of $\mathsf{NICom}_{\mathsf{ccsand}}$ and $\mathsf{ICom}_{\mathsf{ccsand}}$ for some chosen values of u, ℓ are given below. (Here we only use the AND composition. As mentioned above, the OR composition is sometimes more efficient but only under certain restrictions.) The numbers in Tbl. 1 show that the CCS protocol is less efficient than claimed but still more efficient than the previous range proofs.

u	ℓ	$\mathsf{NICom}_{\mathsf{ccsand}}$	$\mathsf{NICom}_{\mathsf{ccsand}}$	Comments
48	5	12 544	38 400	Minimal $\mathsf{NICom}_{\mathsf{ccsand}} + [1, 2] \cdot \mathsf{ICom}_{\mathsf{ccsand}}$
57	5	14 848	38 400	Same parameters as in [CCS08]
633	4	162 304	23 552	Minimal $\mathsf{NICom}_{\mathsf{ccsand}} + 10000 \cdot \mathsf{ICom}_{\mathsf{ccsand}}$, $\mathsf{ICom}_{\mathsf{ccsand}}$

Table 1. Communication of the CCS protocol with some chosen values of u (and implicitly chosen optimal ℓ

Communication of New for some concrete choices of u and ℓ is given in Tbl. 2. Recall that we need to show that $(u-1)(\sigma+1) - 1 \in [0, (u-1)(H'+1)] = [0, 252460800(u-1)]$. We have calculated ℓ according to the point where the recursions of Thm. 1 end, and we note that sometimes its value differs from the predicted value $\lfloor \log_u((u-1)(H+1)) \rfloor$.

u	ℓ	$\mathsf{NICom}_{\mathsf{New}}(u,\ell)$	$\mathsf{ICom}_{\mathsf{New}}(u,\ell)$	Comments
25	6	6 656	27 648	Minimal $\mathsf{NICom}_{\mathsf{New}} + \mathsf{ICom}_{\mathsf{New}}$
48	5	12 544	23 808	Minimal $\mathsf{NICom}_{\mathsf{New}} + \mathsf{ICom}_{\mathsf{New}}$
57	4	14 848	23 808	Same parameters as in [CCS08]
632	4	162 048	16 128	Minimal $\mathsf{NICom}_{\mathsf{New}} + 10000 \cdot \mathsf{ICom}_{\mathsf{New}}$, $\mathsf{ICom}_{\mathsf{New}}$

Table 2. Communication of New with some chosen values of u (and implicitly chosen optimal ℓ

6 Conclusions

We showed that for any H and $1 < u < H$, the interval $[0, H]$ is equal to a sum $\sum G_i * [0, u-1] + [0, H']$, where $0 \leq H' < H$, and both u and ℓ are "small" in terms of H. We gave efficient (closed form) algorithms for computing the coefficients G_i. This result may be interesting by itself in the context of additive combinatorics. We then used this decomposition to show how to derive efficient range proofs for arbitrary intervals $[0, H]$. Compared to the previous work [CCS08], we thus avoided the use of AND composition of Σ-protocols. In addition, (1) we showed also that an earlier result from [LAN02] (that only considered the case $u = 2$) is correct, though it was left unproven in [LAN02], and (2) we pointed out that the range proof from [CCS08] is (in most of the cases) less efficient than claimed there.

References

[BB04] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag.

[BG92] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In Ernest F. Brickell, editor, *CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420, Santa Barbara, California, USA, August 16–20, 1992. Springer-Verlag, 1993.

[Bla97] Kelly Black. Classroom note: Putting constraints in optimization for first-year calculus students. *SIAM Rev.*, 39(2):310–312, 1997.

[Bou00] Fabrice Boudot. Efficient Proofs that a Committed Number Lies in an Interval. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444, Bruges, Belgium, 14–18 May 2000. Springer-Verlag.

[CCS08] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient Protocols for Set Membership and Range Proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 234–252, Melbourne, Australia, December 7–11, 2008. Springer-Verlag.

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient Zero-Knowledge Proofs of Knowledge Without Intractability Assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–373, Melbourne, Victoria, Australia, 18–20 January 2000. Springer-Verlag.

[Che06] Jung Hee Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11, St. Petersburg, Russia, May 28–June 1, 2006. Springer-Verlag.

[CS97] Jan Camenisch and Markus Stadler. Efficient Group Signature Schemes for Large Groups. In Burton S. Kaliski, Jr., editor, *CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, USA, 17–21 August 1997. Springer-Verlag.

[DJ01] Ivan Damgård and Mads Jurik. A Generalisation, A Simplification And Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, Korea, February 13–15, 2001. Springer-Verlag.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A Verifiable Random Function with Short Proofs and Keys. In Serge Vaudenay, editor, *PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[LAN02] 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, March 11–14, 2002. Springer-Verlag.

[Lip03] Helger Lipmaa. On Diophantine Complexity and Statistical Zero-Knowledge Arguments. In Chi Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415, Taipei, Taiwan, November 30–December 4, 2003. Springer-Verlag.

[TV06] Terrence Tao and Van Vu. *Additive Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.