Efficient Protocols for Set Membership and Range Proofs

Rafik Chaabouni

School of Computer and Communication Sciences

Master Project

October 2007

EPFL Supervisor
Prof. Serge Vaudenay
EPFL / LASEC

IBM Supervisors
Dr. Jan Camenisch
Prof. Abhi Shelat
IBM / ZRL

Contents

1 Introduction

The domain of honest verifier zero-knowledge range proof is recognized as being a fundamental aspect in order to build privacy nowadays. Its applications are far from being completely discovered and the research is still in progress.

The goal of this thesis will be to give a major contribution in this domain. In order to do so, we first investigate different cryptographic protocols for proving that a secret lies in some interval, e.g., that the (secret) discrete log of some element y to a base g lies in $[a, b]$ for some integers a and b.

There are some known techniques that address this issue. Depending on the actual size of a and b, some of these are more efficient than others. Moreover, there have been recently new more efficient proposals for specific cases that constitute the current state of the art in this field. We will try to investigate them in order to present a new one more efficient.

The organization of this master thesis will thus start with some introductive notations and definitions in section 2. In section 3, we will recall some basic cryptographic tools necessary for the comprehension of our new set membership and range proof. Section 4 will be dedicated to the current state of the art. In section 5, we describe our contribution to the state of the art, by providing a new efficient set membership, which direct application to range proofs avers to be a major improvement in this domain.

2 Notations and Definitions

2.1 Commitment

One of the basic primitives in cryptography is the notion of *commitment*. A *commiter*, usually named Alice, has a secret x which she wants to commit to. In order to do so, she will hide her secret into a commitment α and release the latter to some *verifier*, named Bob.

Two basic properties characterize the commitment: the hiding and the binding properties. The *hiding property* ensures Alice (the commiter and stakeholder of the secret) that her commitment will leak no information about her secret, i.e. Bob will not gain knowledge of the secret with the commitment only. The *binding property* ensures Bob that upon receiving Alice's commitment, she will not be able to change her mind, her secret which she committed to, i.e. Alice will not be able to cheat on the value of her committed secret.

These two properties can have several order of exactitude. They can be *computationally* achieved, meaning that a cheater will not be able to computationally break the property as he has bounded computational ressources. They can be *statistically* achieved. In this case, even though the cheater has unbounded computational ressources, he cannot achieve his goal statisticly speaking. Last but not least, the properties can be *perfectly* achieved, i.e. there exist no cheater able to break through the property. Note that a commitment cannot be simultaneously perfectly hiding and perfectly binding.

		Commitment properties	
		Hiding	Binding
Level of exactitude	Perfectly	Mutually exclusive events Impossible to find the secret without the commiter revealing it	Impossible to alter the committed secret
	Statistically	Computationally unbounded Cannot find the secret with a high probability	Cannot cheat with a high probability
	Computationally	Computationally bounded Cannot find the secret in a reasonable amount of time	Cannot cheat due to time constraint

Figure 1: Level of exactitude for the *hiding* and *binding* properties

Definition 2.1 (Ensemble) *Let $\mathbb{I}$ be a countable index set. An* ensemble indexed by $\mathbb{I}$ *is a sequence of random variables indexed by $\mathbb{I}$. Namely, any $X = \{X_i\}_{i\in\mathbb{I}}$, where each X_i is a random variable, is an ensemble indexed by $\mathbb{I}$.*

Definition 2.2 (Computational Indistinguishability) *Two ensembles $\{X_w\}_{w\in\mathbb{I}}$ and $\{Y_w\}_{w\in\mathbb{I}}$ with identical index set $\mathbb{I}$ are said to be* computationally indistin-

guishable *if for every probabilistic polynomial-time decision algorithm D, every positive polynomial $p(\cdot)$, and $\exists\ w_0 \in \mathbb{I}$ such that for every $w > w_0$ (here $|w|$ denotes the size of w), we have*

$$|\Pr[D(X_w, w) = 1] - \Pr[D(Y_w, w) = 1]| < \frac{1}{p(|w|)}. \tag{2.1}$$

We denote such sets

$$\{X_w\}_{w\in\mathbb{I}} =_c \{Y_w\}_{w\in\mathbb{I}} \tag{2.2}$$

Definition 2.3 (Statistical Indistinguishability) *Two ensembles $\{X_w\}_{w\in\mathbb{I}}$ and $\{Y_w\}_{w\in\mathbb{I}}$ with identical index set $\mathbb{I}$ are said to be* statistically indistinguishable *if $\exists\ k_0$ such that for every $k > k_0$, every element $a \in \{X_w\}_{w\in\mathbb{I}}$, and every $w \in \mathbb{I}$, we have*

$$\sum_a |\Pr[X_w = a] - \Pr[Y_w = a]| < 2^{-k}. \tag{2.3}$$

We denote such sets

$$\{X_w\}_{w\in\mathbb{I}} =_s \{Y_w\}_{w\in\mathbb{I}} \tag{2.4}$$

Definition 2.4 (Perfect Indistinguishability) *Two ensembles $\{X_w\}_{w\in\mathbb{I}}$ and $\{Y_w\}_{w\in\mathbb{I}}$ with identical index set $\mathbb{I}$ are said to be* perfectly indistinguishable *if for every element $a \in \{X_w\}_{w\in\mathbb{I}}$, and every $w \in \mathbb{I}$, we have*

$$\Pr[X_w = a] = \Pr[Y_w = a]. \tag{2.5}$$

We denote such sets

$$\{X_w\}_{w\in\mathbb{I}} =_p \{Y_w\}_{w\in\mathbb{I}} \tag{2.6}$$

Definition 2.5 (Hiding property) *A commitment scheme is said to be* hiding *if for any two commitment distributions α and α', we have*

$$\{\alpha =_c \alpha'\} \text{ or } \{\alpha =_s \alpha'\} \text{ or } \{\alpha =_p \alpha'\}. \tag{2.7}$$

Furthermore we will precise which type of hiding commitment is achieved, by the type of achieved indistinguishability between the commitment distributions, i.e. computationally, statistically or perfectly hiding commitment scheme.

Definition 2.6 (Binding property) *A commitment scheme is said to be* binding *if for a committed secret x in α, the commiter cannot cheat by opening the commitment α with a distinct secret x'. Formally we distinguish three types of binding, i.e. computationally, statistically or perfectly binding commitment scheme defined as below.*

(i) Perfectly binding *commitment scheme*

A commitment scheme is said to be perfectly binding *if for every commitment procedure $C^\star$ that takes the public parameters Pub_{in} as input and outputs the commitment α together with the opening parameters β_0 and β_1, for two different secret messages respectively m_0 and m_1, in the set of possible messages $\mathbb{M}$, and for the coresponding opening procedure $Open(\alpha, \beta_i, m_i)$ which verifies that β_i correctly opens the secret message m_i, we have*

$$\Pr\left[\begin{array}{l|l} \forall i \in \{0,1\},\ \forall m_i \in \mathbb{M}, & m_0 \neq m_1, \\ Open(\alpha, \beta_i, m_i) = true & (\alpha, m_0, m_1) \leftarrow C^\star(Pub_{in}) \end{array}\right] = 0. \tag{2.8}$$

(ii) Statistically binding *commitment scheme*

A commitment scheme is said to be statistically binding *if* $\exists\ k_0$ *such that for every* $k > k_0$*, every commitment procedure* $C^\star$ *that takes the public parameters* Pub_{in} *as input and outputs the commitment* α *together with the opening parameters* β_m *(for every secret message* m *in the set of possible messages* $\mathbb{M}$*), and for the coresponding opening procedure* $Open(\alpha, \beta_i, m_i)$ *which verifies that* β_i *correctly opens the secret message* m_i*, we have*

$$\Pr\left[\begin{array}{l|l} \forall i \in \{0,1\},\ \forall m_i \in \mathbb{M}, & m_0 \neq m_1, \\ Open(\alpha, \beta_i, m_i) = true & (\alpha, m_0, m_1) \leftarrow C^\star(Pub_{in}) \end{array}\right] \leqslant 2^{-k}. \tag{2.9}$$

(iii) Computationally binding *commitment scheme*

A commitment scheme is said to be computationally binding *if* $\exists\ w_0 \in \mathbb{I}$ *such that for every* $w > w_0$ *(here* $|w|$ *denotes the size of* w*), every positive polynomial* $p(\cdot)$*, every probabilistic polynomial-time commitment procedure* $C^\star$ *that takes the public parameters* Pub_{in} *as input and outputs the commitment* α *together with the opening parameters* β_m *(for every secret message* m *in the set of possible messages* $\mathbb{M}$*), and for the coresponding opening procedure* $Open(\alpha, \beta_i, m_i)$ *which verifies that* β_i *correctly opens the secret message* m_i*, we have*

$$\Pr\left[\begin{array}{l|l} \forall i \in \{0,1\},\ \forall m_i \in \mathbb{M}, & m_0 \neq m_1, \\ Open(\alpha, \beta_i, m_i) = true & (\alpha, m_0, m_1) \leftarrow C^\star(Pub_{in}) \end{array}\right] < \frac{1}{p(|w|)}. \tag{2.10}$$

Let p be a prime number and let G be the multiplicative group of integers modulus p. Let g, h be two generators for G. A standard method to cryptographically commit to a secret value x is to randomly choose a value $r \in_R [1, p-1]$ and compute $\alpha := g^x h^r (\mathrm{mod}\ p)$. The commitment α is perfectly hiding and computationally binding (assuming that the commiter does not choose g and h but is rather given these values).

2.2 Proofs

Once a commitment has been made, a verifier would want some guarantees on the committed secret. He will want to be assured that the committed secret obeys to some properties.

To begin with, Bob will want to be assured that the commitment makes sense and that Alice knows the secret. In other words he wants to be sure that Alice can open the commitment. He might also want to verify that the secret obeys some computational properties. For instance the commiter might have to prove that the secret is a positive integer, a square number, a product of two primes, etc.

In all these issues, the commiter, who will also be called the *prover*, will have to provide along with the commitment a *proof of knowledge* to the verifier. This proof will be denoted as $PK(x : predicates\ on\ x)$ and will have some properties. The major property these proofs will have is the *zero-knowledge* aspect. Of course Alice does not want to give more knowledge to Bob than what he is supposed to learn. Then comes the *completeness* property as some

proof might not always succeed in order to maintain the zero-knowledge. The *soundness* of the proof is also a crutial element of the proof as it tells Bob how much Alice can cheat with the proof. These proves will thus be given with some properties (e.g. length of the protocol) and with some parameters (e.g. range of the values).

Definition 2.7 (Zero-knowledge) *A proof is said to be* zero-knowledge *if for any transcript t_{PV} between the prover and the verifier, there exist a simulated transcript t_{Sim}, from an algorithm simulator $Sim(\cdot)$, such that*

$$\{t_{PV} =_p t_{Sim}\} \text{ or } \{t_{PV} =_s t_{Sim}\} \text{ or } \{t_{PV} =_c t_{Sim}\}. \tag{2.11}$$

Furthermore we will precise which type of zero-knowledge proof is achieved, by the type of achieved indistinguishability between the transcript distributions, i.e. perfect, statistical or computational zero-knowledge proof. Note that we are assuming the honest verifier model and that there exist transformation techniques to generalize it for any verifier (see [8]).

Definition 2.8 (Soundness) *The* soundness *of a proof of membership coresponds to the probability that the verifier accepts a false proof. Formally for some predicates $p(\cdot)$ on messages $m \in \mathbb{M}$ defining a language set $L_p = \{\alpha \mid \alpha = E(m, r) \wedge p(m)\}$, where the function E computes the commitment from a random seed r and the secret message m, we define the soundness for a pair of interactive machines (P, V) (for any P, and where V has a polynomial time complexity) as such:*

$$\Pr\left[(P, V)(x) = \text{“}accept\text{”}\right], \forall\, x \notin L_p. \tag{2.12}$$

For proofs of knowledge, the notion of soundness is decomposed into four major notions:

- *ϵ_{view,P^*}: probability of P^* to make V accept conditionned on the $view$,*
- *the knowledge error $\kappa(k)$: allowed threshold for ϵ_{0,P^*}, i.e. threshold for P^* on how much he can make V accept while knowing nothing about w,*
- *the advantage of P^*, $Adv_{\kappa,M,p}(P^*, k)$: probability that "M fails" on the $view$ generated by the "experiment",*
- *the failure probability $\nu(k)$: threshold for the advantage of P^*.*

Definition 2.9 (Completeness) *The* completeness *of a proof corespond to the probability of the proof completion. Formally for some predicates $p(\cdot)$ on messages $m \in \mathbb{M}$ defining a language set $L_p = \{\alpha \mid \alpha = E(m, r) \wedge p(m)\}$, where the function E computes the commitment from a random seed r and the secret message m, we define the completeness for a pair of interactive machines (P, V) (for any P, and where V has a polynomial time complexity) as such:*

$$\Pr\left[(P, V)(x) = \text{“}accept\text{”}\right], \forall\, x \in L_p. \tag{2.13}$$

The most basic proofs of knowledge are called Σ-protocols and intend to be honest-verifier zero-knowledge. They consist of three major steps: commitment, challenge and response (see figure 2). The main objective of this proof is to persuade the verifier on the existence of some specific binary relation R between a common input v and a witness w to the later. Our focus will be to

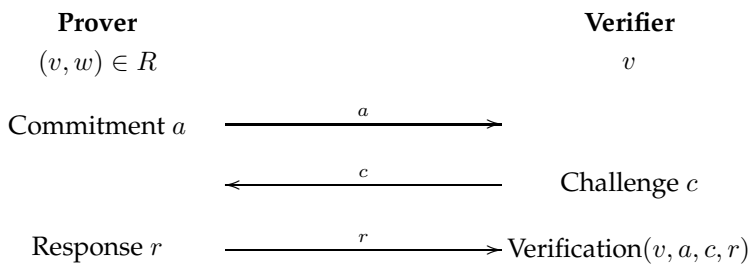


Figure 2: **Σ-protocols.**

consider the secret x contained in a given commitment as a part of the witness, and the commitment α as a part of the common input. R is therefore a subset of $\{0,1\}^* \times \{0,1\}^*$ such that if $(v, w) \in R$, then the length of w is bounded by $p(|v|)$, for some given polynomial $p(\cdot)$.

Two major properties accompany Σ-protocols: the *special honest-verifier zero-knowledge* and the *special soundness property*. The first property states the existence of a polynomial-time simulator M, which outputs transcripts with the same probability distribution as in a real honest conversation. Moreover M is not based on the knowledge of the witness w, but rather the common input v. The second property emphasis the potential recovery of a witness w for the common input v, for any v and any pair coresponding valid transcript with two different challenge. A complete formalization of Σ-protocols and there properties can be found in [6], where they were first introduced.

Suppose Alice publishes a commitment α and now wishes to prove to Bob that the secret value which has been committed in α lies in some range $[a, b]$. Moreover, Alice would like this proof to be a zero-knowledge proof in which nothing other than the truth of this range is conveyed in the proof. General results in zero-knowledge show that such a proof, called range proof, is possible.

Definition 2.10 (Expansion rate) *For a zero-knowledge proof where the secret committed value x is chosen by the prover in $\mathbb{I}$, and is proved to the verifier to belong to the range $\mathbb{J}$, we define the expansion rate δ as*

$$\delta := \frac{|\mathbb{J}|}{|\mathbb{I}|} \tag{2.14}$$

where $|\mathbb{J}|$ and $|\mathbb{I}|$ means the number of elements of respectively the set $\mathbb{J}$ and the set $\mathbb{I}$.

Our goal, however, is to find an efficient method to make this proof. In a composite group, the four-squares method is one way to show any range proof. This method seems to be efficient since the amount of work and the soundness error of the proof are independent of a and b. The basic idea is to reduce the problem of showing membership in an interval to proving that a committed value is "positive." (In a composite group, the committer does not know the order of the group, and so cannot make negative values wrap around.) In order to do this, one finds four values $x_1, \ldots, x_4$ such that $x_1^2 + x_2^2 + x_3^2 + x_4^2 = x$. Finding such four values is possible for any positive x (this follows from Lagrange's Four-Square Theorem also known as Bachet's conjecture).

The four-squares method, alas, does not work for prime-order groups. However, the same technique of showing that a committed value is "positive" can be applied, albeit somewhat less efficiently. In the case of a prime order group, showing that a value is "positive" amounts to showing that it is less than $p - 1/2$. Essentially the idea is to commit to each bit of the secret and show that these bit commitments together represent the same value that was committed in α. With these bit commitments, one can string together a statement such as "the top bit is zero or (the top bit is one and the second bit is zero) or ....". The size of this proof is related to the number of bits in p (typically a few hundred) which makes this technique inefficient.

A few other approaches to this problem have been suggested by Schoenmakers in [17] and in [19]. Hence, in order to have a better appreciation of our contribution we will first explain some tools that we used, such as the oblivious transfer in [4], and then we will dive in the current state of the art.

3 Tools for the State of the Art

Before digging into the realm of zero-knowledge range proofs, we will first present usefull tools. These ones will help us in the understanding of the state of the art.

To begin with, we will recall the Schnorr proof as the basic sigma protocol. Then we will study the commitment scheme presented by Ivan Damgård and Eiichiro Fujisaki in [7], together with a proof of knowledge and a multiplicative proof. We will also present the oblivious transfer presented by Jan Camenisch, Gregory Neve and Abhi Shelat in [4] as it will contribute in the understanding of our new efficient set membership.

3.1 Schnorr proof

Schnorr proofs are the first efficient Σ-protocols proposed, even though they were not introduced as such in [16]. Nevertheless, this proof has become a fundamental tool for knowledge proofs. The aim of it, is to attest the knowledge of a discrete logarithm, using cyclic groups (see figure 3). Let g be a generator of the cyclic group $\mathbb{Z}_p$, where p is a prime number. Our secret x will be the discrete logarithm of some number h. Thus the common input of this Σ-protocol is composed by g, $\mathbb{Z}_p$ and h. The witness is the secret value x. The relation between the witness and the common input is dictated by the discrete logarithm problem.

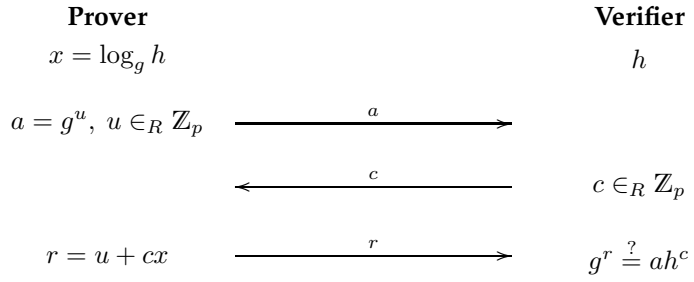


Figure 3: **Schnorr proof.**

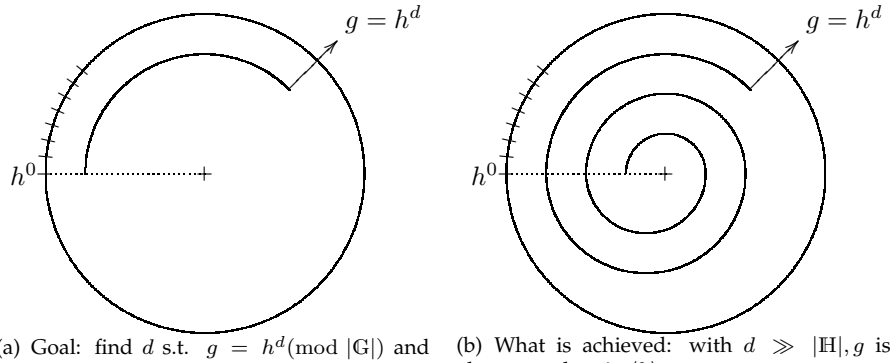
3.2 Damgård and Fujisaki scheme

3.2.1 Commitment scheme

Damgård and Fujisaki presented there commitment scheme in three major parts: a set-up, a commit and an open phase (see figure 5).

In the set-up phase the verifier first runs a probabilistic polynomial time algorithm $\mathcal{G}$ with a k bit input of ones (denoted 1^k) to obtain the description of a set $\mathbb{G}$. This set is built such that it is composed of two distinctive subgroups, a large one with only large prime factors (denoted $\mathbb{H}$) and a small one with small prime factors (denoted $\mathbb{U}$). Thus if we take a random element h in $\mathbb{G}$, it will have a high probability of being in the subgroup $\mathbb{H}$. We can furthermore verify this by checking if $h^{|\mathbb{U}|} \not\equiv 1 (\mathrm{mod}\ |\mathbb{G}|)$. Secondly the verifier chooses h as

described above ($\in_R \mathbb{G}, \in \mathbb{H}$), and will try to find a g such that $g \in_R \langle h \rangle$ (where $\langle h \rangle$ denotes the set generated by h).



(a) Goal: find d s.t. $g = h^d \pmod{|\mathbb{G}|}$ and $g \in_R \langle h \rangle$

(b) What is achieved: with $d \gg |\mathbb{H}|, g$ is almost random in $\langle h \rangle$

Figure 4: Goal vs. what is achieved

However the description of $\mathbb{G}$ does not contain the factorization of the order of $\mathbb{G}$ (hence the factorization of the order of $\mathbb{H}$ is unknown). In order to find a correct g, the verifier will simply raise h to some large random power d, much larger than the actual order of h (see figure 4). Ideally we would like d to be such that the probability of outputting a given g is bellow than 2^{-k} and randomly chosen in $\langle h \rangle$. Hence ideally d should be picked randomly in the set $[0, 2^{2B+k}]$ where $B = \log_2(|\langle h \rangle|)$. With this constraint we obtain that $2^{2B+k} = 2^k \cdot (|\langle h \rangle|^2)$ which ensures our goal for g. However as we do not know $|\langle h \rangle|$, we will use $|\mathbb{H}|$ instead.

The verifier finishes the set-up phase by sending to the prover the description of $\mathbb{G}$, g, h, and by proving him that $g \in \langle h \rangle$[1].

The commit phase consists only for the prover to choose a random number r, to compute the commitment $\alpha \equiv g^x h^r \in \mathbb{G}$, and to send α to the verifier.

In order to open the commitment, the prover needs to reveal his secret x, his choice of r and a value μ such that: $\alpha \equiv \mu g^x h^r \in \mathbb{G}$ and $\mu^{|\mathbb{U}|} = 1$.

According to theorem 1 in [7], the above commitment scheme is perfectly hiding and computationally binding.

3.2.2 Knowledge proof

The knowledge proof (described in figure 6) proves to the verifier that the prover knows the secret x committed in a given commitment α, with an expansion rate of $\delta = C(k) \cdot (2^k + 2)$, where $C(k)$ denotes a super-polynomial function in k, as suggested in [7].

First let us assume that the prover chooses x in some publicly known range $[-T, T]$, and commits to his secret by computing and sending $\alpha = g^x h^r$ as seen above (with $r \in_R \left[0, 2^{B+k}\right]$). In order to run through the proof, he will need to pick two variables y and s that will help him hide his secret and his random

[1] The description of $PK((g,h) : g \in \langle h \rangle)$ can be found in detail in section A.1 of [4]

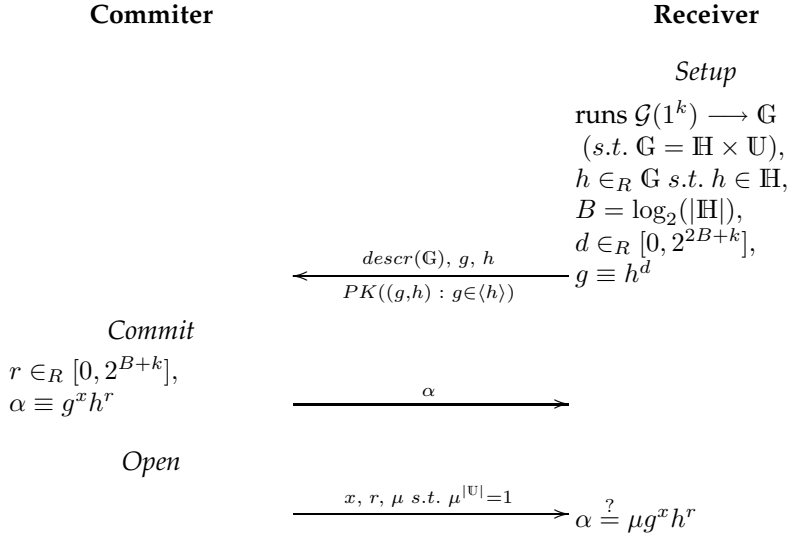


Figure 5: **Damgård and Fujisaki commitment scheme.**

number chosen for the commitment. He will thus compute $d \equiv g^y h^s$ and send d to the verifier who will reply by a random number e taken from the set $[0, C(k)[$. This last set can be contested as the value $e = 0$ allows a cheating prover to use any secret x outside the expected range. A better choice of set would have been to use the set $]0, C(k)]$. Finally the prover simply computes $u = y + xe$ and $v = s + re$, sends them to the verifier who will check the equivalence between $g^u h^v$ and $d\alpha^e$, along with the constraint that $u \in \left]-TC(k),\ TC(k)(2^k + 1)\right[$.

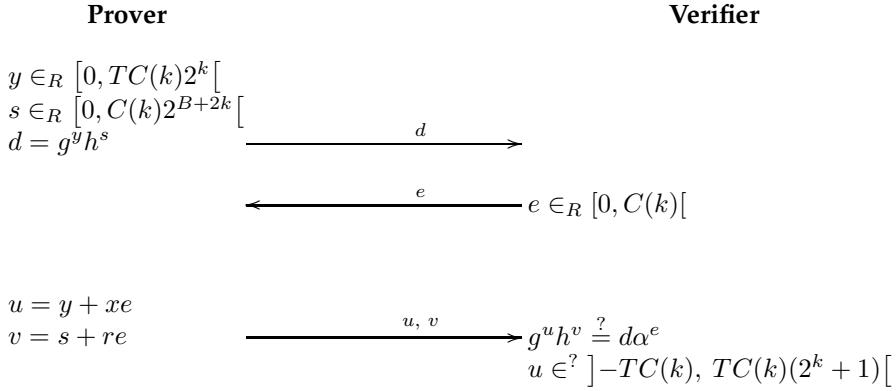


Figure 6: **Damgård and Fujisaki proof of knowledge.**

Now that the proof process is described, let us see in more detail why we get the mentioned expansion rate of $\delta = C(k) \cdot (2^k + 2)$. For a cheating prover, his goal will be to send a u and v such that the verifier will accept them. In that

regard, the prover will try to guess the value of e. We will suppose the general case where $e \in [e_1, e_2] \subseteq [0, C(k)[$ and $y \in [y_1, y_2]$. We have thus, according to the value of x:

$$x \geqslant 0 \;\Rightarrow\; (y + xe) \in [y_1,\; y_2 + xe_2] \subseteq \left]-TC(k),\; TC(k)(2^k + 1)\right[\tag{3.15}$$

$$x \leqslant 0 \;\Rightarrow\; (y + xe) \in [y_1 + xe_2,\; y_2] \subseteq \left]-TC(k),\; TC(k)(2^k + 1)\right[\tag{3.16}$$

We thus see by taking successively the value $-TC(k)$ and the value $TC(k)(2^k + 1)$ for y that the secret x will be bounded by:

$$\frac{-TC(k)(2^k + 2)}{e_2} < x < \frac{TC(k)(2^k + 2)}{e_2} \tag{3.17}$$

Thus the largest set containing x that the verifier can expect is

$$x \in \left]-TC(k)(2^k + 2),\; TC(k)(2^k + 2)\right[\tag{3.18}$$

which explains the described expansion rate.

Objective: $PK((x, r) : \alpha = E(x, r) \;\wedge\; x \in \left]-TC(k)(2^k + 2),\; TC(k)(2^k + 2)\right[)$
(expansion rate: $\delta = C(k)(2^k + 2)$)

Soundness: According to theorem 2 in [7], this proof of knowledge has a knowledge error $\kappa(k)$ in $\mathcal{O}\left(1/C(k)\right)$ and a failure probability $\nu(k) = 9\epsilon(k)l(k)$, where $\epsilon(k)$ represents the maximal probability with which any adversary bounded in time by $t(k)$ breaks the root problem (see [7] for more precise details and definitions).

Completeness: The protocol finishes after three exchanged messages.

3.2.3 Multiplicative proof

As mentionned before, we can have proofs for predicates on the secret. Here the proof will consist of convincing the verifier that the secret x_3 committed in α_3 is actually the product of secrets x_1 and x_2, committed respectively in α_1 and α_2, i.e. $x_3 = x_1 x_2$ (see figure 7). As previously, the prover is expected to chose his secrets in the set $[-T, T]$ and to follow section 3.2.1 for the commitment procedure. We have thus the following reasoning on α_3:

$$\begin{aligned} \alpha_3 &= g^{x_3} h^{r_3} = g^{(x_1 x_2)} h^{r_3} \\ &= g^{x_1 x_2} \cdot h^{(r_1 x_2 - r_1 x_2) + r_3} \\ &= (g^{x_1} h^{r_1})^{x_2} \cdot h^{r_3 - r_1 x_2} \\ &= \alpha_1^{x_2} \cdot h^{r_3 - r_1 x_2} \end{aligned} \tag{3.19}$$

In order to prove that $x_3 = x_1 x_2$, the prover has first to convince the verifier that he knows both x_1 and x_2, then prove the veracity of the relation 3.19. The proof of knowledge for x_1 and x_2 is identical to section 3.2.2.

We can notice that the relation 3.19 is similar to $\alpha = g^x h^r$ where g has been replaced by α_1, where the secret message is x_2, and the random number

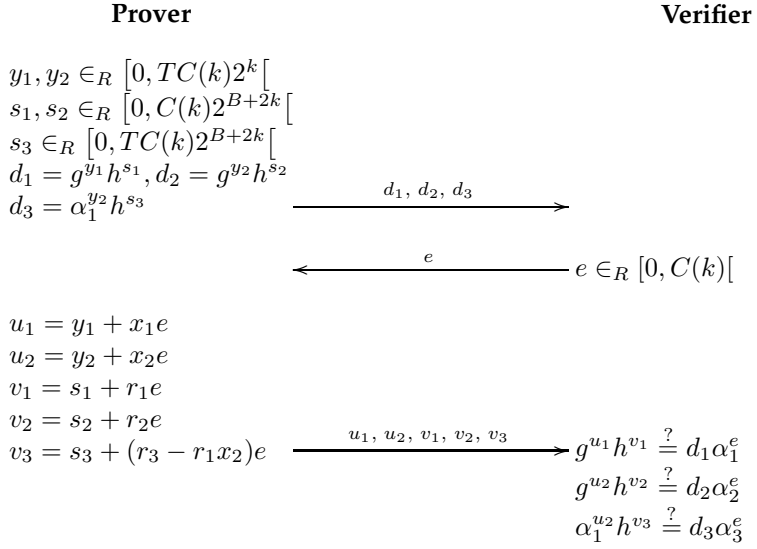


Figure 7: **Damgård and Fujisaki multiplicative proof.**

is $(r_3 - r_1 x_2)$. Thus we can use the previous proof of knowledge replacing the coresponding parts as mentionned, in order to prove the relation 3.19 and ultimately to achieve the required goal.

Note that the range of s_3 has to be different than the one for $s_1,\ s_2$ as the random number range that needs to be hidden is different. Indeed, in the previous proof of knowledge we have $r \in_R \left[0, 2^{B+k}\right]$, where as for the last part of the multiplicative proof, the random number that we are considering is $(r_3 - r_1 x_2) \in \left[-T2^{B+k},\ (T+1)2^{B+k}\right]$. As the upper bound for the random number is multiplied by $T+1$, it is natural to apply this scaling to s_3. The lower bound s_3 should ideally conserve this scaling, but for efficiency concerns we can limit s_3 to positive values (this will reduce the probability of having a negative v_3 and thus to have to compute the inverse of h at the verifier side).

3.3 Oblivious transfer

Oblivious transfers have become an important primitive in cryptography. These are protocols for information retrival designed in such a way that the information provider cannot learn which information is requested, and the information retriever does not gain any more knowledge than what he asked for. These primitives have been consistently studied and several variants have been found. One of which is the adaptive variant from Jan Camenisch, Gregory Neve and Abhi Shelat in [4]. They gave two solutions for the k out of N oblivious transfer which consists of a receiver queriying for k elements out of N messages provided by the sender. We will only focus on their new protocol for the standard model (see figure 8).

The general idea behind it, is for the sender to hide the actual messages M_i in a value B_i using some verifiable random elements A_i and his secret key h.

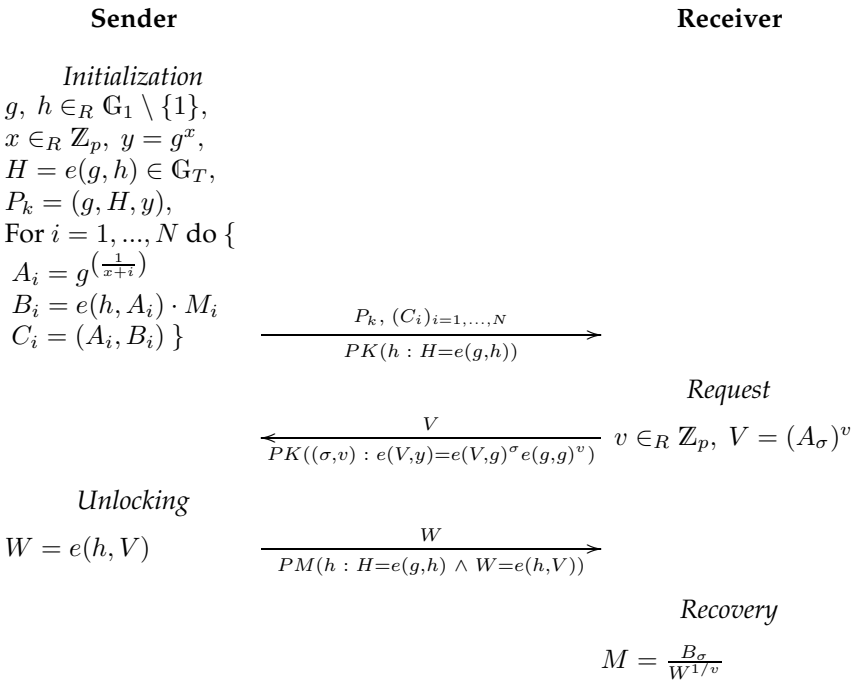


Figure 8: k out of N Adaptive Oblivious Transfer in the standard model from [4]. A complete descritpion of the proof of knowledge $PK((\sigma, v) : e(V, y) = e(V, g)^{\sigma} e(g, g)^{v})$ as well as the proof of membership $PM(h : H = e(g, h) \wedge W = e(h, V))$ can be also found in [4] respectively at section A.2 and A.3.

The values B_i and A_i will be available to the receiver. In order to retrieve the desired message M_i, the receiver will first blind his choice by raising the value A_i to the power of his secret key v and assigning this value to V. Thus, only with this last value, the sender is not able to determine which message is requested. However he has the possibility to provide the unlocking element W of B_i whithout revealing his secret unlocking all messages. This is feasible by making use of a bilinear map. The operations done on V will affect A_i without knowing the value of i.

4 (Prior) State of the art

A large amount of research has already been achieved in the domain of honest verifier zero-knowledge range proofs. In order to be able to find a new efficient solution, one has first to fully understand the previous accomplished work. This will be the purpose of this section. We will not try to present all the possible solutions but rather focus on the relevant ones.

We will thus cover the work done by Boudot in [3] and Lipmaa (in [13] and in [14]) on the sum of four square method, followed by the work accomplished by Berry Schoenmakers in his presentations [17] and [18].

4.1 Boudot's range proof

In order to prove that a given integer is contained in a given range, Boudot has proposed two schemes depending if tolerance in the expansion rate is allowed, or if an exact proof is required. In both schemes the proof is divided into two positivity proofs. The novelty introduced by Boudot resides in these positivity proofs.

If a small tolerance is allowed, the positive secret is decomposed as the sum of the highest square possible and the remaining small positive value. The square will be proved as being a correct square and thus a positive element, while a Chan-Frankel-Tsiounis proof (see [5]) will be used for the remaining value. This is described in figure 9.

If an exact proof is required, Boudot simply proposed to artificially enlarge the original set such as the level of uncertainty is hidden by the integer constraint of the secret value, as shown in figure 10.

Note that Boudot uses t, l, and s as security parameters, taking values respectively 80, 40 and 40.

4.2 Lipmaa's range proof

Lipmaa followed the positivity property of square numbers and proposed to prove positivity by decomposing the secret number into the sum of four square (see [13] and [14]). This decomposition was first introduced by Lagrange in 1770 and a probabilistic polynomial time algorithm was given by Rabin and Shallit in 1986 (see [15]). Thus this decomposition is realistic and has been converted into a honest verifier zero-knowledge proof system for positive intigers (see Protocol 2 in [13]).

Let us also mention the contribution of Jens Groth in [12] where he pushed further the positivity proofs with square decomposition down to a three square method.

4.3 Berry Schoenmakers' scheme

Berry Schoenmakers focused his work on exact proofs for small intervals. Instead of trying to prove range proof through a square decomposition, he based his research on the study of bit decomposition. Each bit of the secret is proven to be eather 0 or 1 with a variant of Schnorr proof. This yields to a proof of knowledge of the from $PK\left(x, r : C = g^r h^x \wedge x \in [0, 2^k)\right)$ as described in his presentations [17] and [18].

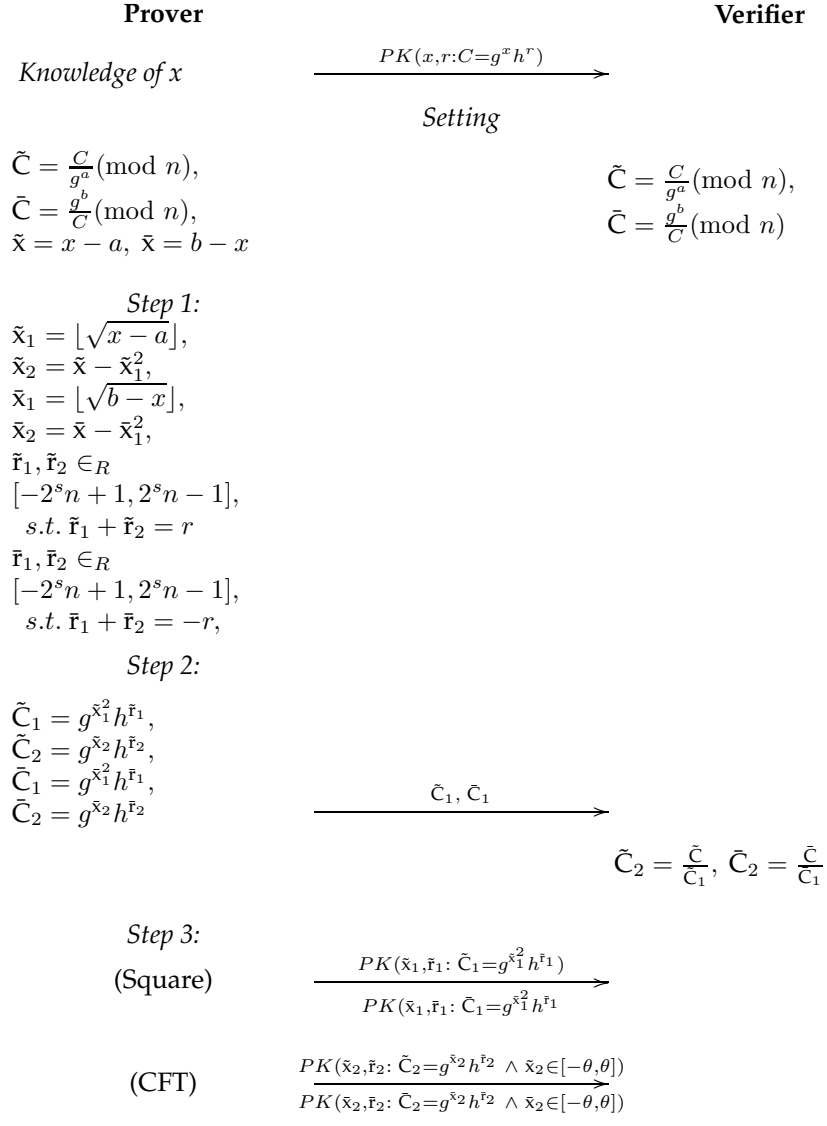


Figure 9: $PK(x, r : C = g^x h^r \wedge x \in [a - \theta, b + \theta])$ with $\delta = 1 + \varepsilon$.

However an arbitrary upper bound can be used by decomposing the set $[0, L)$ with an AND-composition or an OR-composition. Let us assume that $2^{k-1} < L \leqslant 2^k$:

$$[0, L) = [0, 2^k) \cap [L - 2^k, L) \tag{4.20}$$
$$[0, L) = [0, 2^{k-1}) \cup [L - 2^{k-1}, L) \tag{4.21}$$

The subsets for the AND-composition being both of size 2^k, the amount of work required becomes equivalent to $4k$ Schnorr proofs. Similarly for the OR-composition, the equivalent of $4(k-1)$ Schnorr proofs are required. Note that

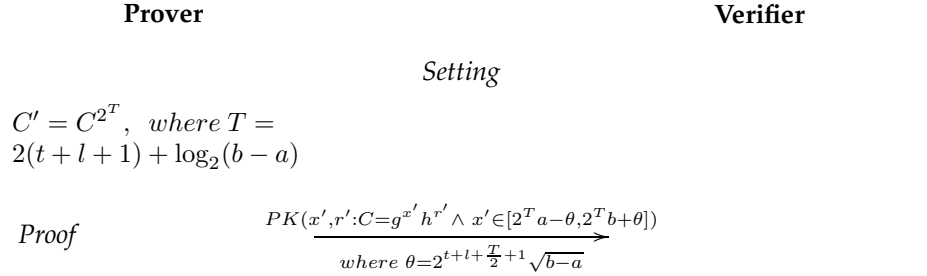


Figure 10: $PK(x, r : C = g^x h^r \wedge x \in [a, b])$ where $x' = 2^T x$.

in order to prove that a given number is contained in a range of size 2^k we can simply add or substract the constant noise parameter to the secret value in order to fall back to a proof in $[0, 2^k)$, for instance:

$$x \in [L - 2^k, L) \Leftrightarrow (x - L + 2^k) \in [0, 2^k). \tag{4.22}$$

From here, Berry Schoenmakers pushed forward the idea to find other means to express the upper bound in order reduce the work needed. For instance if we have $2^{k_1} < L < 2^{k_1+1}$ we can write $L = 2^{k_1} + R$ such that $2^{k_2-1} < R \leqslant 2^{k_2}$ with $k_2 \leqslant k_1$. The OR-composition will thus result in the following:

$$[0, L) \quad = \quad [0, 2^{k_1}) \cup [L - 2^{k_2}, L) \tag{4.23}$$

with an amount of work of $2(k_1 + k_2)$ Schnorr proofs.

The two major approaches that he proposed are to consider L as either a product or a sum of two numbers. By doing this scheme recursively he can decrease the amount of work needed. However the overall communication load will still be in $O(k)$.

The product case is possible whenever we can write the upper bound as the product $L = ab$. This will result on the unique decomposition of the secret as $x = vb + y$ such that $0 \leqslant v < a$ and $0 \leqslant y < b$. Three commitments will be required for this matter: $C = g^r h^x$, $A = g^{s_1} h^v$ and $B = g^{s_2} h^y$, such that $a = s_1 b + s_2 (\mathrm{mod}\ p)$, where $s_1, s_2 \in_R \mathbb{Z}_p$. The result of this construction will be $C = A^b B$. Therefore if we prove that the subsecret element v is in $[0, a)$ and that the subsecret element y is in $[0, b)$, we end up proving that x is indeed in $[0, L)$.

The sum case is somehow similar to the previous one, in the sense that we decompose x as being either in $[0, a)$ with the help of the commitment C, or either in $[a, a + b)$ with the help of the commitment $\frac{C}{h^a}$ which will be used to prove that the coresponding secret $(x - a)$ lies in $[0, b)$.

When combined recursively these two approaches suggest an ammount of work in terms of Schnorr proofs, equal to the complexity of the range size. We recall that the complexity of a number L is defined as the minimal number of element 1 in order to write L with products and sums of element 1, including parentheses. For instance we have $7 = (1 + 1) * (1 + 1 + 1) + 1$. The complexity

function of natural numbers, that we will call $W(\cdot)$, is defined by the sequence A005245 in the "On-Line Encyclopedia of Integer Sequences". To resume we can consider that the achievement of Schoenmakers in terms of communication load is bounded by $W(L)$ Schnorr proofs, where L is the size of the range considered.

5 New efficient protocols for set membership and range proofs

5.1 Introduction

As mentionned previously, the current state of the art is bounded by a communication load of $O(k)$. In order to do a major improvement in the field of honest verifier zero-knowledge set membership and range proof, one has to find a protocol that is asymptotically more efficient. This improvement rises with the work done by Schoenmakers combined with the knowledge of oblivious transfers. Indeed, instead of limitting ourselves to a bit decomposition, we consider a u-ary decomposition, where our basis is no longer Schnorr proofs, but rather a signature-based set membership inspired from the presented adaptive oblivious transfer.

5.2 New set membership protocol

We persent here our new solution inspired by Oblivious Transfer, that we will call Set Membership from Oblivious Transfer (SMOT, see figure 11). The goal is to proove to some verifier that our secret is contained in some public set Φ. In order to do so the verifier signs every value contained in Φ and publishes these signatures. Thus, the prover receives a signature on the particular element σ to which C is a commitment. This step represent the initialization of our new solution. The verifier will simply have to "blinds" this received signature and perform a proof that he knows a signature for his committed sercret value. The proof then does not depend on the size of the secret itself but rather on the cardinality of the set Φ from wich the secret is picked from. The novelty of this approach is that the first verifier message can be re-used in other proofs of membership; indeed, we use this property to achieve our results for range proofs.

5.2.1 Computational assumptions.

Our protocol require bilinear groups and associated hardness assumptions. Let PG be a pairing group generator that on input 1^k outputs descriptions of multiplicative groups $\mathbb{G}_1, \mathbb{G}_T$ 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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, 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 5.1 (Strong Diffie-Hellman Assumption [2]) *We say that the q-Strong Diffie-Hellman assumption associated to a pairing generator PG holds if for all probabilistic polynomial-time adversaries A, the probability that $A(g, g^x, \ldots, g^{x^q})$ where $(\mathbb{G}_1, \mathbb{G}_T) \leftarrow \mathsf{PG}(1^k)$, $g \leftarrow \mathbb{G}_1^*$ and $x \leftarrow \mathbb{Z}_p$, outputs a pair $(c, g^{1/(x+c)})$ where $c \in \mathbb{Z}_p$ in negligible in k.*

5.2.2 Boneh-Boyen signatures.

Our scheme relies on the elegant Boneh-Boyen short signature [2] 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 [9].

Security under a 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 [2] which we paraphrase below:

Lemma 5.2 ([2](Lemma 3.2)) *Suppose the q-Strong Diffie Hellman assumption holds in $(\mathbb{G}_1, \mathbb{G}_T)$. Then the basic Boneh-Boyen signature scheme is q-secure against an existential forgery under a weak chosen message attack.*

Theorem 5.3 *If the $|\Phi|$-Strong Diffie-Hellman Assumption associated with a pairing generator* PG *holds, then protocol in figure 11 is a zero-knowledge argument of set membership for a set Φ.*

Proof: The completeness of the protocol follows by inspection. The soundness follows from the extraction property of the proof of knowledge and the unforgeability of the random function. In particular, the extraction property implies that for any prover P^* that convinces V with probability ϵ, there exists an extractor which interacts with P^* and outputs a witness (σ, ρ, η) with probability $poly(\epsilon)$. If $\sigma \notin \Phi$, then P^* can be (almost) directly be used to mount a weak chosen-message attack against the Boneh-Boyen signature scheme with probability $poly(\epsilon)$ of succeeding. Thus, ϵ must be negligible.

Finally, to prove honest verifier zero-knowledge, follow the initialization and the blinding instructions honestly (using a random $\sigma \in \Phi$ to compute V). Then run the simulator for the Σ-protocol in the following steps. Since G_1 is a prime-order group, then the blinding is perfect in the first step; thus the zero-knowledge property follows from the zero-knowledge property of the Sigma protocol.

The resulting overall communication load is given by

$$Com_l(\Phi) \quad = \quad |\Phi| \cdot |\mathbb{G}_1| + (5 \cdot |\mathbb{G}_1| + 3 \cdot |\mathbb{G}_T| + 5 \cdot |\mathbb{Z}_p|) \tag{5.24}$$

5.3 Application to range proof

The previous protocol can be applied for range proof if the set Φ contains consecutive elements. However this will not be efficient for large ranges. In order to make it efficient in this later case, we focus on the decomposition of our secret σ to the base u ($\Phi = \mathbb{Z}_u$). This decomposition needs to be done carefully

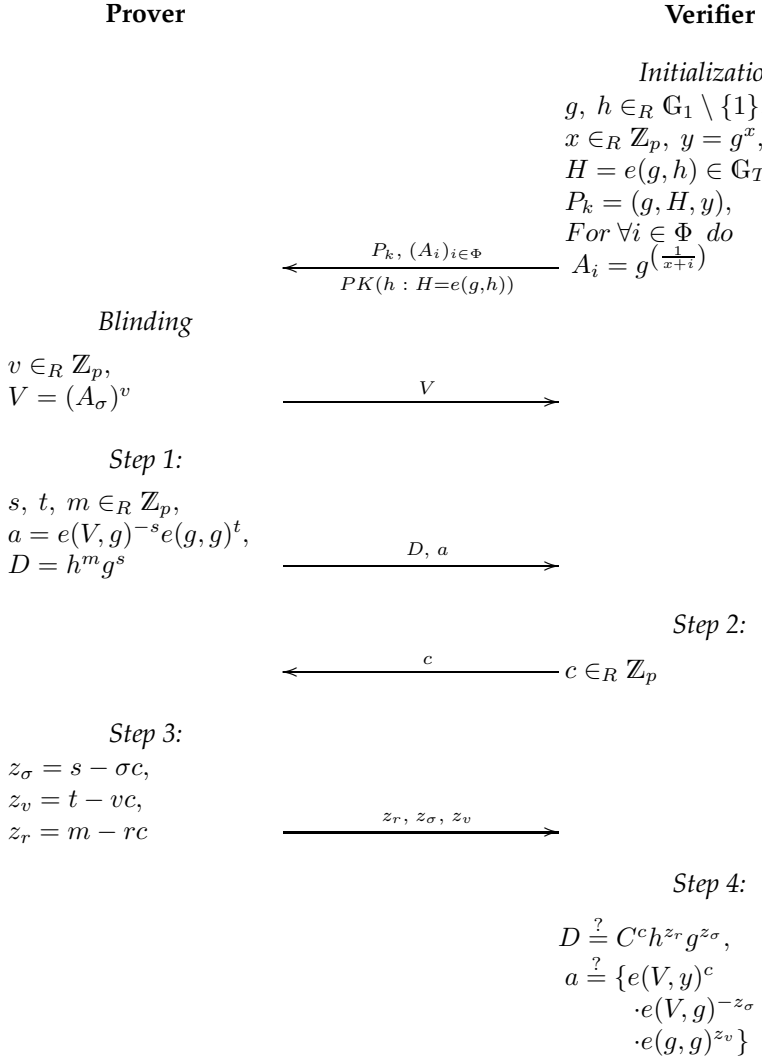


Figure 11: **SMOT scheme.** $PK(\sigma : \sigma \in \Phi)$ with commitment $C = g^\sigma h^r$.

as on how to choose u. Assuming that $\sigma \in [0\ B)$, the goal is to minimize the communication load under the constraint $u^k \geqslant B$.

First let us present how to prove that our secret σ lies in $[0\ u^k)$, then we will review the communication complexity. We will present at this stage the asymptotic analysis and explain the issues for concrete optimizations. In a third step, we will explain how to handle any arbitrary range $[a, b)$ for the range proof. We will thus finish with a concrete example in order to show a practical comparison.

5.3.1 Range proofs from our new set membership protocol

Similarly to the previous protocol, we will name this new one Range Proof from Oblivious Transfer (RPOT - see figure 12). If we look at the decomposition of σ in the base u, we will obtain k elements as such: $\sigma = \sum_j (\sigma_j u^j)$. Hence we apply on each of the decomposition coefficient a SMOT proof with $D = \prod_j (g_j^{s_j} h^m)$. This modification of the D value enables us to take into account that each σ_j is a coefficient of the σ decomposition.

Lemma 5.4 *If the* $\log k$*-Strong Diffie Hellman assumption associated to a pairing generator* $\mathsf{PG}(1^k)$ *holds, there exists a zero-knowledge range argument for the range* $[0, u^\ell)$ *where* $u^\ell < \{0,1\}^{k-1}$.

Proof: Completeness follows from inspection. As before, the soundness follows from the unforgeability of the Boneh-Boyen signature and the extraction property of the proof of knowledge protocol. The honest-verifier zero-knowledge property follows from the perfect blinding of the signatures in the first phase, and the corresponding honest-verifier zero-knowledge property of the Σ-protocol.

5.3.2 Communication Complexity

The first message of the protocol consists of u signatures, the public key P_k sent by the verifier to the prover and a proof of knowledge $PK(h : H = e(g,h))$. The prover then sends ℓ blinded values back. Thus, the first phase requires

$$Init_l(u,\ell) \quad = (u+\ell+3)\cdot|\mathbb{G}_1| + 2\cdot|\mathbb{G}_T| + |\mathbb{Z}_p| \tag{5.25}$$

communication. The second phase of the protocol involves a proof of knowledge. The prover sends $\ell+1$ first-messages of a Σ-protocol. The verifier sends a single challenge, and the prover responds with $2\ell+1$ elements. Thus the overall communication load according to the parameters u and ℓ is:

$$\begin{aligned} Com_l(u,k) \quad = \quad & u\cdot|\mathbb{G}_1| + \ell\cdot(|\mathbb{G}_1| + |\mathbb{G}_T| + 2\cdot|\mathbb{Z}_p|) \\ & +(4\cdot|\mathbb{G}_1| + 2\cdot|\mathbb{G}_T| + 3\cdot|\mathbb{Z}_p|) \end{aligned} \tag{5.26}$$

Finding the optimal u and ℓ thus involves solving

$$\min c_1 u + c_2\ell + c_3 \tag{5.27}$$

$$\text{s.t. } u^\ell \geqslant B \tag{5.28}$$

Notice that the bit-committing protocol corresponds to a setting where $u = 2$ and $\ell = k$ which leads to a total communication complexity $O(k)$. Since our protocol allows us to choose more suitable u, we first show that the asymptotic complexity of our approach is smaller than the prior protocols.

Asymptotic Analysis For the asymptotic analysis, we may ignore the constants c_1, c_2 and c_3. Moreover, we can take $B \approx p/2$ as this is sufficient for showing that a committed value is "positive," i.e., in the range $[0\ (p-1/2)]$. Since $p \approx 2^k$, the constraint becomes $u^\ell \geqslant 2^{k-1}$.

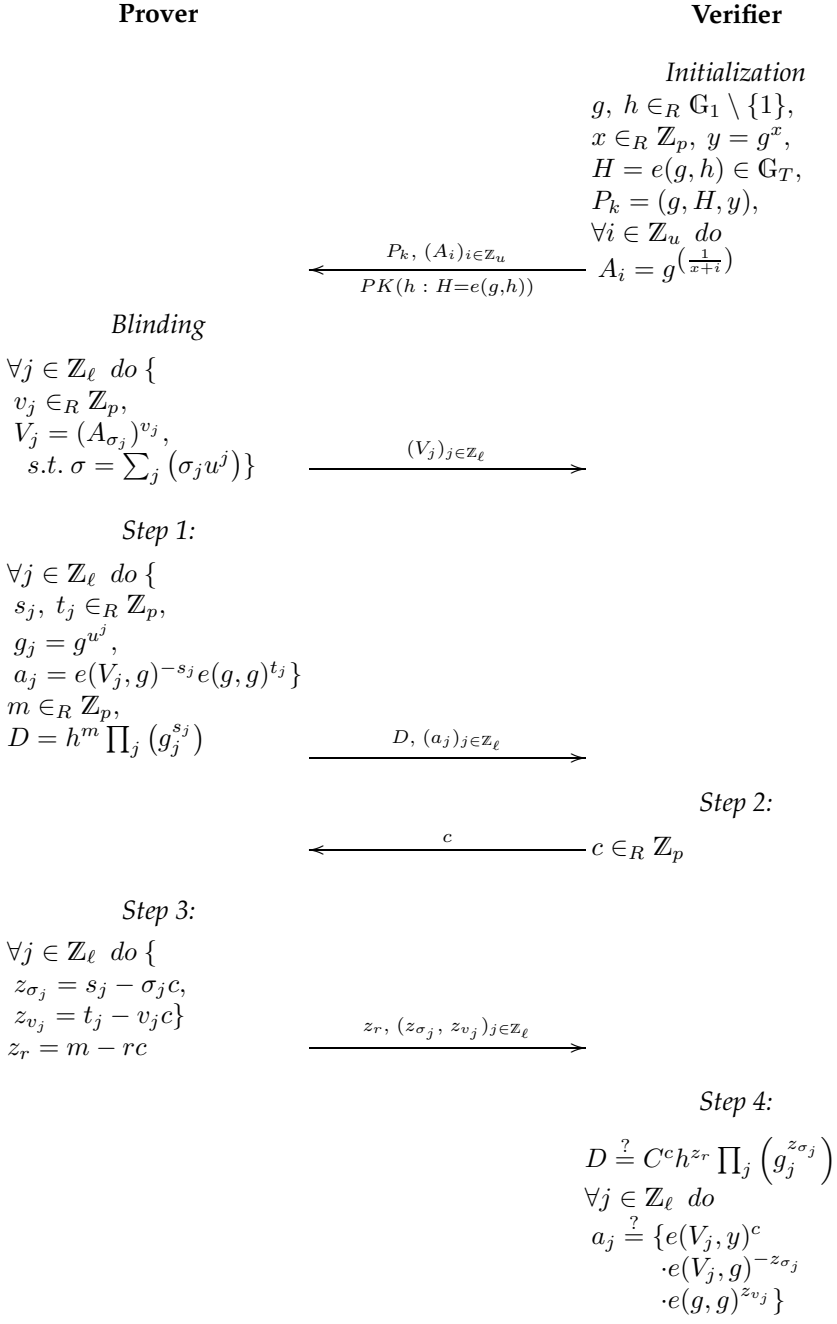


Figure 12: **RPOT scheme.** $PK(\sigma : \sigma \in [0\ u^\ell))$ with commitment $C = g^\sigma h^r$.

By taking logs and dividing, we have that $\ell \approx \frac{k}{\log u}$. Setting $u = \frac{k}{\log k}$ then

we get that

$$u = O\left(\frac{k}{\log k}\right), \tag{5.29}$$

$$\ell = O\left(\frac{k}{\log k - \log\log k}\right) \tag{5.30}$$

resulting in a total communication complexity of

$$Com(u,\ell) = O\left(\frac{k}{\log k - \log\log k}\right) \tag{5.31}$$

which is asymptotically smaller than $O(k)$.

Concrete Optimization Not only is our solution asymptotically better, but it also performs well for realistic concrete parameters. In order to perform the optimization for concrete parameters we substitute the constraint that $u^\ell \approx B$ into the equation $u + \ell$ above. To minimize, we set the derivative with respect to u to 0 and attempt to solve the equation:

$$c_1 - \frac{c_2 \log B}{u \log^2 u} = 0 \tag{5.32}$$

which simplifies to

$$u \log^2 u = \frac{c_2 \log B}{c_1}. \tag{5.33}$$

This equation cannot be solved analytically. However, given B, c_1 and c_2, we can use numerical methods to find a good u as described in [1].

5.3.3 Handling Arbitrary Ranges $[a, b)$

The above protocol works for the range $[0, u^\ell)$. In order to handle an arbitrary range $[a, b]$, we use an improvement of a folklore reduction described by Schoenmakers in [17] and [18]. Suppose that $u^{\ell-1} < b < u^\ell$. To show the $\sigma \in [a, b)$, it suffices to show that

$$\sigma \in [a, a + u^\ell) \cap \sigma \in [b - u^\ell, b) \tag{5.34}$$

Proving that our secret lies in both subsets can be derived from our general proof that $\sigma \in [0, u^\ell)$:

$$\sigma \in [b - u^\ell, b) \iff \sigma - b + u^\ell \in [0, u^\ell) \tag{5.35}$$

$$\sigma \in [a, a + u^\ell) \iff \sigma - a \in [0, u^\ell). \tag{5.36}$$

Note that the u signatures and the verification key need to be sent only once for both subsets. Since both a, b are public, the only modification necessary is the verifier's check, which should now be:

$$D \stackrel{?}{=} C^c g^{-B+u^\ell} h^{z_r} \prod_j \left(g_j^{z_{\sigma_j}}\right), \tag{5.37}$$

$$D \stackrel{?}{=} C^c g^{-A} h^{z_r} \prod_j \left(g_j^{z_{\sigma_j}}\right). \tag{5.38}$$

Thus, $\ell \cdot (|\mathbb{G}_T| + 2 \cdot |\mathbb{Z}_p|) + (|\mathbb{G}_1| + 2|\mathbb{Z}_p|)$ extra elements are sent in the protocol.

This scheme can be further optimized when $A + u^{\ell-1} < B$ with an OR-composition. Indeed, the decomposition becomes:

$$[A, B) = [B - u^{\ell-1}, B) \cup [A, A + u^{\ell-1}). \tag{5.39}$$

The needed modifications are similar to the previous case; the efficiency arises from the fact that we are now working with $\mathbb{Z}_{\ell-1}$. The length of the range set can also be optimized. Indeed if $B - A = u^\ell$ then the proof reduces to proving that $\sigma - A \in [0, u^\ell)$.

Combining this analysis with Lemma 5.4 yields the following theorem.

Theorem 5.5 *If the* $\log k$*-Strong Diffie Hellman assumption associated to a pairing generator* $\mathsf{PG}(1^k)$ *holds, there exists a zero-knowledge range argument for the range* $[a, b]$ *where* $0 < a < b < \{0,1\}^{k-1}$ *whose communication complexity is* $O(\frac{k}{\log k - \log\log k})$.

5.3.4 Concrete Example

Concretely, if we pick $B = 599644800$ (which will represent people born before 1989, with their birth date encoded using the Unix Epoch system), we can find the optimal values of u and ℓ by either computing them numerically or by following [1]. Both methods will lead us to $u = 57$ and $\ell = 5$, which minimize the overall communication load:

$$Com_l(57, 5) = 66 \cdot |\mathbb{G}_1| + 7 \cdot |\mathbb{G}_T| + 13 \cdot |\mathbb{Z}_p| \tag{5.40}$$

which includes an initialization load of

$$Init_l(57, 5) = 65 \cdot |\mathbb{G}_1| + 2 \cdot |\mathbb{G}_T| + \cdot |\mathbb{Z}_p| \tag{5.41}$$

Let us illustrate this optimization case with a concrete example. We will assume that an airline company wants to provide special offers to its young clients from a third party. However the exact age of clients should not be divulged to the third party. This offer targets those who are born between 1981 and 1989 (not included). Following the previous example, the birth date will be a secret number between $[347184000, 599644800)$. Here the best option will be to use the OR-composition as $A + u^{\ell-1} < B$ (we know from the previous example that $u = 57$ and $\ell = 5$). Using parameters from Galbraith, Paterson, and Smart [10], we estimate 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. This leads to an overall communication load of[2]:

$$\begin{aligned} Com_{l\cup}(u = 57, \ell = 5) &= (u + \ell + 1) \cdot |\mathbb{G}_1| \\ &\quad + (2\ell - 2) \cdot |\mathbb{G}_T| + 4\ell \cdot |\mathbb{Z}_p| \\ &= 45824\, bits \end{aligned} \tag{5.42}$$

In order to have a better appreciation of this result, we compare it to previous protocols in figure 13.

[2]Note that we are here considering the case where the common input to both prover and verifier is already composed by g, h, u, ℓ, and the commitment C, which means that they already ran $PK(h : H = e(g, h))$

Scheme	*Communication Complexity*
Our new range proof	45824 bits
Boudot's method	48946 bits
Standard bit-by-bit method	96768 bits
Schoenmakers' method	50176 bits

Figure 13: Communication load comparison for range proof $[347184000, 599644800)$

6 Conclusion

The initial objectives of this master thesis where considerably high at the beggining, considering that we were aiming for a new state of the art in the domain of honest verifier zero-knowledge set membership and range proofs. In order to cope with this requirements, we first studied some initial background on the domain and tried to get in touch with the newest development in this area of research. An important part of the time has been spent on the study and precise analysis of the curent flow of research, especially the work done by Berry Schoenmakers. From this built experience, we managed to have a glimpse on a potential unexplored path. This was made possible thanks to the advise of Dr. Jan Camenisch and Prof. Abhi Shelat, who has actually opened a new faculty in the University of VirginiaTech. However we had still to materialize this path into real protocols. This work has been now fulfilled as a patent will be deposit on our results, together with an attempt to publish it. The knowledge gained through this experience enabled us to realize our initial objectives with serenity.

However we only grasped one of the fruits of knowledge. Future research is still required in order to reduce as much as possible the communication load and the amount of work for these proofs. Even in our own results, we did not look in detail the potentiality to use or to modify our protocol for revocation of anonymous credentials, group membership proof or even set membership for certified attribute. Some research should be done in order to learn what would be the minimum load that we can reach in order to see the boundary of the remaining research. Potential clues for this boundary could be given by the domain of information theory.

References

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

[2] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[3] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT*, pages 431–444, 2000.

[4] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 2007.

[5] Agnes Hui Chan, Yair Frankel, and Yiannis Tsiounis. Easy come - easy go divisible cash. In *EUROCRYPT*, pages 561–575, 1998.

[6] Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocol*. PhD thesis, University of Amsterdam, 1997.

[7] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2002.

[8] Ivan Damgard, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs dishonest verifier in public cain zero-knowledge proofs. In *CRYPTO*, number Theory, pages 325–338, 1995.

[9] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *Public Key Cryptography*, pages 416–431, 2005.

[10] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006.

[11] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, New York, NY, USA, 2000.

[12] Jens Groth. Non-interactive zero-knowledge arguments for voting. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 3531 of *Lecture Notes in Computer Science*, pages 467–482, 2005.

[13] Helger Lipmaa. Statistical zero-knowledge proofs from diophantine equations. Cryptology ePrint Archive, Report 2001/086, 2001.

[14] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In Chi-Sung Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415. Springer, 2003.

[15] M. O. Rabin and J. O. Shallit. Randomized algorithms in number theory. *Comm. Pure Appl. Math.*, 39(S):239–256, 1986.

[16] Claus P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.

[17] Berry Schoenmakers. Some efficient zeroknowledge proof techniques. Monte Verita, March 2001.

[18] Berry Schoenmakers. Interval proofs revisited. Milan, Italy, September 2005.

[19] Berry Schoenmakers. Private communication. 2006.

Index