Non-interactive Distributed-Verifier Proofs and Proving Relations among Commitments

Masayuki Abe[1], Ronald Cramer[2], and Serge Fehr[2]

[1] NTT Laboratories, Japan
abe@isl.ntt.co.jp
[2] BRICS*, Aarhus University, Denmark
{cramer,fehr}@brics.dk

Abstract. A *commitment multiplication proof*, CMP for short, allows a player who is committed to secrets s, s' and $s'' = s \cdot s'$, to prove, without revealing s, s' or s'', that indeed $s'' = ss'$. CMP is an important building block for secure general multi-party computation as well as threshold cryptography.

In the standard cryptographic model, a CMP is typically done interactively using zero-knowledge protocols. In the random oracle model it can be done non-interactively by removing interaction using the Fiat-Shamir heuristic. An alternative non-interactive solution in the distributed setting, where at most a certain fraction of the verifiers are malicious, was presented in [1] for Pedersen's discrete log based commitment scheme. This CMP essentially consists of a few invocations of Pedersen's verifiable secret sharing scheme (VSS) and is secure in the standard model.

In the first part of this paper, we improve that CMP by arguing that a building block used in its construction in fact already constitutes a CMP. This not only leads to a simplified exposition, but also saves on the required number of invocations of Pedersen's VSS. Next we show how to construct non-interactive proofs of partial knowledge [8] in this distributed setting. This allows for instance to prove non-interactively the knowledge of ℓ out of m given secrets, without revealing which ones. We also show how to construct efficient non-interactive zero-knowledge proofs for circuit satisfiability in the distributed setting.

In the second part, we investigate generalizations to other homomorphic commitment schemes, and show that on the negative side, Pedersen's VSS cannot be generalized to arbitrary (black-box) homomorphic commitment schemes, while on the positive side, commitment schemes based on q-one-way-group-homomorphism [7], which cover wide range of currently used schemes, suffice.

1 Introduction

Commitment schemes play an important role as a primitive in cryptographic protocols. Applications are found for instance in the construction of zero-knowledge

* Basic Research in Computer Science (www.brics.dk), funded by the Danish National Research Foundation.

proofs and arguments, secure multi-party computation and threshold cryptography. Using a commitment scheme, a player can commit to a secret value s by publishing a *commitment* C, in such a way that the commitment C reveals nothing about the secret s, i.e., the scheme is *hiding*. The player can later *open* C to reveal s in a way verifiable by everyone else, i.e., it is *binding* in the sense that the player can't open C to any other value than s.

Many protocols using commitments require a player at some point to prove certain relations among a set of committed values, without revealing these committed values in the process. Assuming that addition or multiplication of secret values is well-defined, a player committed to s, s' and s'' will typically be required to prove that $s'' = s + s'$ or that $s'' = ss'$. If the commitment scheme is homomorphic, as is the case with many known commitment schemes, the additive relation is trivial to handle, even non-interactively. A *commitment multiplication proof (CMP)*, i.e., a secure protocol to handle the multiplicative relation, is generally less trivial to design.

In the two-player setting, there exist efficient *interactive* zero-knowledge protocols for all known homomorphic schemes [7]. These protocols can be adapted in a natural way to a distributed setting with n players and where up to t of them are malicious, for instance by simply letting each of the players be engaged in a separate run of the two-player protocol with the prover.

In [1] it is shown how this approach can be substantially improved by taking advantage of the fact that sufficiently many players are guaranteed to be honest. Namely, it is shown how to handle CMP in this distributed setting *non-interactively* in the case of Pedersen's discrete logarithm based commitment scheme. This CMP essentially consists of a few Pedersen VSS's and is non-interactive (from the prover's point of view) in case everyone plays honestly, while the prover might have to answer accusations otherwise. We call this *non-interactive with accusing*. Moreover, it is totally non-interactive if $t < n/3$.

In the first part of this paper, we improve that CMP by arguing that a building block used in its construction in fact already constitutes a CMP. This not only leads to a simplified exposition, but also saves on the required number of invocations of Pedersen's VSS. Next we show a new technique to construct non-interactive proofs of partial knowledge in this distributed setting, thereby extending the results of [8] for the interactive two-player case. This allows for instance to prove non-interactively the knowledge of ℓ out of m given secrets, without revealing which ones. As an application, it allows to make the proof of correctness of a ballot in the [10] voting scheme non-interactive without resorting to random oracles. We also show how to construct efficient non-interactive zero-knowledge proofs for circuit satisfiability in the distributed setting.

In the second part, we investigate generalizations to other homomorphic commitment schemes, and show that on the negative side, Pedersen's VSS cannot be generalized to arbitrary (black-box) homomorphic commitment schemes, while on the positive side, commitment schemes based on q-one-way-group-homomorphism [7], which cover wide range of currently used schemes, suffice. Finally, we show how this positive result leads to error-free non-interactive zero-knowledge proofs of membership for non-trivial languages in this distributed setting.

We proceed by repeating the concepts of commitment schemes and (verifiable) secret sharing and by recalling the concrete schemes of Pedersen in the following Section 2. In Section 3 we define (zero-knowledge) distributed-verifier proofs and we show that Pedersen's VSS can be seen as such a proof, and in Section 4 we present the CMP protocol and the proof protocols for partial knowledge and for general circuit satisfiability. Finally, in Section 5, we investigate to what extent the above protocols can be generalized to other homomorphic commitment schemes.

2 Preliminaries

2.1 Pedersen's Commitment Scheme

A *commitment scheme* of the kind we consider over a finite domain $\mathcal{S}$ is given by a function family

$$\mathsf{com}_{pk} : \mathcal{S} \times \mathcal{R}_{pk} \to \mathcal{C}_{pk}$$

indexed by a *public key* pk, where $\mathcal{R}_{pk}$ and $\mathcal{C}_{pk}$ are finite sets. In a set-up phase, a concrete public key pk and thus function com_{pk} is fixed in a prescribed manner. By publishing a *commitment* $C = \mathsf{com}_{pk}(s, r)$ for a random $r \in \mathcal{R}_{pk}$, such a scheme allows a party, Alice, to *commit* herself to a secret $s \in \mathcal{S}$, such that the commitment C reveals nothing about the secret s (*hiding property*) while on the other hand Alice can *open* C to s by publishing (s, r) *but only to* s (*binding property*).

If $\mathcal{S}$ is a field K (or, more generally, a ring), then such a commitment scheme is called *homomorphic*, if the following holds: For any commitments C and C' and any number $\lambda \in K$, one can compute commitments S and P such that being able to open C and C' to values s and s', respectively, allows to open S to the sum $s + s'$ and P to the product λs.

A well known homomorphic commitment scheme is the Pedersen commitment scheme [5, 2, 16], given by

$$\begin{aligned} \mathsf{com}_{g,h} : \mathbb{F}_q \times \mathbb{F}_q &\to \quad G \\ (s, r) \quad &\mapsto g^s h^r \end{aligned}$$

where q is a prime, G is a (multiplicative) group of order $|G| = q$ in which computing discrete logarithms is (assumed to be) hard, e.g. a subgroup of $\mathbb{F}_p^*$, and g and h are randomly chosen generators of G. This scheme is unconditionally hiding and computationally binding, and it is homomorphic: If $C = \mathsf{com}_{g,h}(s, r)$ and $C' = \mathsf{com}_{g,h}(s', r')$ then $C \cdot C' = \mathsf{com}_{g,h}(s + s', r + r')$ and $C^\lambda = \mathsf{com}_{g,h}(\lambda s, \lambda r)$.

2.2 Pedersen's Verifiable Secret Sharing Scheme

In a *secret sharing scheme* a *dealer* distributes a *secret* s to n players $P_1, \ldots, P_n$ (for simplicity we set $P_i = i$) by privately sending to each player P_i a *share* s_i

in such a way that, for a fixed *threshold* t, up to t players have no information about the secret s (*privacy*) while $t+1$ players (or more) are able to reconstruct it (*correctness*). While secret sharing only guarantees security against curious players that try to gather information they are not supposed to obtain but otherwise behave honestly, its stronger version *verifiable secret sharing* [11], VSS for short, is secure in the following sense against up to t dishonest players and a possibly dishonest dealer that behave in an arbitrary manner.

Privacy: In case of an honest dealer, the information the dishonest players gain during the distribution of the secret s gives no information about s.

Correctness: As soon as the distribution is completed, there exists a fixed value s' such that every honest player will output s' as a result of the reconstruction, and if the dealer is honest, then $s' = s$.

The Pedersen VSS scheme [16] is based on Shamir's secret sharing scheme [17] and Pedersen's commitment scheme.

Protocol $\mathsf{Share}_{g,h}$

1. To share a secret $s \in \mathbb{F}_q$, the dealer chooses a random polynomial $f_s(X) = a_0 + a_1 X + \ldots + a_t X^t \in \mathbb{F}_q[X]$ of degree at most t with constant coefficient $a_0 = s$, and he commits himself to this *sharing polynomial* $f_s(X)$ by broadcasting commitments $A_0, \ldots, A_t$ of $a_0, \ldots, a_t$, respectively. For every player P_i, the dealer computes the share
$$s_i = f_s(i) = s + a_1 i + \ldots a_t i^t \in \mathbb{F}_q$$
and he opens the corresponding commitment
$$C_i = A_0 \cdot A_1^i \cdot \ldots \cdot A_t^{i^t}$$
privately to P_i, using the homomorphic property of the commitment scheme.
2. If P_i does not accept the opening, then P_i broadcasts an accusation against the dealer.
3. To any accusation of a player P_i, the dealer responds by opening C_i publicly.
4. If he fails to do this correctly then the sharing is rejected, otherwise it is accepted.

After the execution of this protocol, assumed that it has been accepted, every player P_i is committed to his share s_i by the commitment C_i, and he holds the corresponding information to open it. Hence, the reconstruction works as follows.

Protocol $\mathsf{Reconstruct}_{g,h}$

Every player P_i publicly opens C_i to s_i. The shares s_i that have been correctly opened are then taken to reconstruct the secret by interpolation.

The pair ($\mathsf{Share}, \mathsf{Reconstruct}$) is a VSS if (and only if) $t < n/2$. Privacy holds unconditionally while correctness holds under the assumption that computing discrete logarithms is hard.
The scheme can be made completely non-interactive from the dealer's point of view in case $t < n/3$ by replacing the steps 3. and 4. by

3.' If the number of accusations is larger than t, then the sharing is rejected, otherwise it is accepted.

Namely, in this case, if the sharing is accepted then there are at least $t+1$ honest players that have not accused the dealer in step 2. and hence have a consistent sharing that allows to reconstruct the secret (see also the proof of Proposition 1).

Remark. Consider an accepted execution of the sharing protocol. By correctness, a secret value s' is fixed that can later be reconstructed. Since the information used by the players in the reconstruction originated with the dealer, we can conclude that the dealer knows this secret. In fact, it is straight forward to show, as we do later on, that the dealer not only knows s' but he also knows how to open the commitment A_0 used in the sharing protocol (to s').

3 Distributed Verifier Proofs

3.1 Model and Definition

We consider a *prover* P who wants to prove to a set of n *verifiers* $\mathcal{V} = \{V_1, \ldots, V_n\}$, that he knows some witness w without revealing it. We assume an adversary that can actively corrupt up to t of the n verifiers as well as the prover P, where we consider both cases $t < n/2$ and $t < n/3$. Some of the protocols require the adversary to be computationally bounded, and we assume him to be *static*, meaning that he has to corrupt the parties *before* the protocol execution. We assume that secure pairwise channels as well as broadcast channels are either provided by cryptographic means (in case of a bounded adversary) or given as primitives, though, for simplicity, also in the former case we will treat them as being perfectly secure.

Consider now two sets $\mathcal{W}$ and $\mathcal{I}$ and an efficiently verifiable relation $R \subseteq \mathcal{W} \times \mathcal{I}$. Given some *public information* $I \in \mathcal{I}$, the prover wants to convince the verifiers that he knows a *witness* $w \in \mathcal{W}$ with $(w, I) \in R$.

Definition 1. *A* distributed verifier proof (of knowledge) *for relation R is a protocol among a prover P and n verifiers $V_1, \ldots, V_n$ (all polynomially bounded), with a common input I, a private input w by P and a public output accept or reject, such that the following security properties hold, even if up to t of the n verifiers as well as the prover might be corrupted by the adversary.*

Correctness: *If P is honest and $(w, I) \in R$, then the output will be accept.*
Soundness: *There exists a knowledge extractor that can efficiently compute from the joint view of the honest players a witness w' satisfying $(w', I) \in R$, assumed that the output of the protocol is accept.*

This soundness condition can come in three flavors: perfect*,* unconditional*, or* computational*. Meaning that the condition holds with probability 1, with overwhelming probability, or under some computational assumption, respectively.*

A distributed verifier proof is called non-interactive*, if the structure of the protocol is as follows. The prover sends to every verifier one message, a personal* partial proof*, and then every verifier votes to either accept or reject the proof, depending on whether he accepts or rejects his partial proof, and the outcome of the protocol is accept if and only if not more than t verifiers vote for rejection. It is called* non-interactive with accusing*, if it is non-interactive except that in case there are some rejections, the prover must broadcast the corresponding partial proofs, and the outcome of the protocol is accept if and only if none of these published proofs is rejected.*

Finally, it is called zero-knowledge*, if the adversary can simulate his view of the protocol.*

The above soundness condition highlights the power of the distributed verifier setting in two ways: 1) The prover is *not* given to the knowledge extractor as a *rewindable* black-box. Thus, no rewinding argument is needed to prove the soundness of a protocol. 2) In case of *perfect* soundness it asserts that there is no knowledge error. Hence, acceptance of the proof always implies the knowledge of a witness w'. Of course, one can relax the definition by allowing to rewind the prover so that it becomes seamless with the standard definition of proof of knowledge [4] with a single verifier.

Such a distributed verifier proof can also be seen as a proof of membership where the prover proves the *existence* of a witness w with $(w, I) \in R$ and therefore that I belongs to the language $L_R = \{I \mid \exists\, w : (w, I) \in R\}$. A proof of membership for language L in this model can be defined similarly, with the corresponding correctness and soundness conditions as follows.

Correctness: *If P is honest and $x \in L$, then the output will be accept.*
Soundness: *If the output of the protocol is accept, then $x \in L$.*

Again, soundness can come in different flavours. It is, however, important to note that in a usual single verifier proof perfect soundness can be achieved only for trivial languages while this is not true for distributed verifier proofs. This will be addressed further in Section 5.3.
Proofs of membership in a distributed setting have also been introduced in [3] under the name of *network zero-knowledge proofs*.

3.2 Pedersen's VSS as a Distributed Verifier Proof

Let $\mathsf{com}_{g,h} : \mathbb{F}_q \times \mathbb{F}_q \to G$, $(s, r) \mapsto g^s h^r$ be the Pedersen commitment scheme. For a commitment $C = \mathsf{com}_{g,h}(s, r)$ let $\mathsf{Proof}_{g,h}(C)$ denote an execution of $\mathsf{Share}_{g,h}$ with secret s, except that in step 1. of the protocol, $A_0 = C$ is taken as commitment of $a_0 = s$. Then, $\mathsf{Proof}_{g,h}(C)$ is a zero-knowledge proof that the dealer can open the commitment C. More formally, for relation

$$R_{g,h} = \{((s, r), C) \mid s, r \in \mathbb{F}_q, C = \mathsf{com}_{g,h}(s, r)\} \subseteq (\mathbb{F}_q)^2 \times G$$

we have

Proposition 1. *Protocol* $\mathsf{Proof}_{g,h}$ *is a perfectly sound zero-knowledge distributed-verifier proof for relation* $R_{g,h}$*, non-interactive in case* $t < n/3$ *and non-interactive with accusing in case* $t < n/2$.

We stress that we have soundness and zero-knowledge independent of the quality of the commitment scheme $\mathsf{com}_{g,h}$. In fact, this holds even in case the discrete logarithm $\log_g h$ is known and hence the binding property does not hold at all.

Proof of Proposition 1: Since, to any possible accusation, the honest prover only broadcasts correct information, the proof will be accepted. It remains to show soundness and zero-knowledge.
Soundness: Assume that the proof has been accepted, and let A be the set of honest players, respectively, in case of $t < n/3$, the set of honest players who have not accused the dealer. In any case, $|A| \geq t+1$ (and we assume without loss of generality that $A = \{1, \ldots, t+1\}$) and every player $P_i \in A$ can open his commitment C_i to, say, s'_i. Let $\lambda_1, \ldots, \lambda_{t+1}$ be the *reconstruction coefficients* for the players in A. That is, $\sum_{i=1}^{t+1} \lambda_i s_i = s$ for correctly computed shares s_i of s, which means that $\sum_{i=1}^{t+1} \lambda_i \sum_{k=0}^{t} a_k i^k = s = a_0$ and hence $\sum_{i=1}^{t+1}(\lambda_i i^k) = \delta_{k0}$ (where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise). Because of the homomorphic property of the commitment scheme, the players of A can open the commitment $C' = \prod_{i=1}^{t+1} C_i^{\lambda_i}$ to $s' = \sum_{i=1}^{t+1} \lambda_i s'_i$. However, as

$$C' = \prod_{i=1}^{t+1} C_i^{\lambda_i} = \prod_{i=1}^{t+1} \left(\prod_{k=0}^{t} A_k^{i^k} \right)^{\lambda_i} = \prod_{k=0}^{t} A_k^{\sum_i (\lambda_i i^k)} = A_0 = C \qquad (1)$$

it follows that they can open C (to s').
Zero-knowledge: Let A be the set of corrupted players. We assume without loss of generality that $A = \{1, \ldots, t\}$. We make use of the well known fact that from the secret s and the shares $s_1, \ldots, s_t$ of the players in A, all the random sharing coefficients $a_1, \ldots, a_t$ can be computed in a linear way. Hence, writing $s_0 = s = a_0$, for every $k \in \{0, \ldots, t\}$ there exist coefficients $\mu_{k0}, \ldots, \mu_{kt}$ such that $a_k = \sum_{j=0}^{t} \mu_{kj} s_j$, which means that $s_i = \sum_{k=0}^{t} a_i i^k = \sum_{k=0}^{t} \sum_{j=0}^{t} \mu_{kj} s_j i^k$ and hence $\sum_{k=0}^{t} \mu_{kj} i^k = \delta_{ij}$.
Given the commitment C for s, the players in A can simulate their view of the protocol as follows. For every $P_i \in A$ they choose $s_i \in \mathbb{F}_q$ at random and compute a (random) commitment C_i for s_i, and for $k = 0, \ldots, t$ they compute $A_k = \prod_{j=0}^{t} C_j^{\mu_{kj}}$, where $C_0 = C$, such that $A_0 = C$ and for every $i \in A$

$$\prod_{k=0}^{t} A_k^{i^k} = \prod_{k=0}^{t} \left(\prod_{j=0}^{t} C_j^{\mu_{kj}} \right)^{i^k} = \prod_{j=0}^{t} C_j^{\sum_k (\mu_{kj} i^k)} = C_i \qquad (2)$$

Finally, it is not hard to see that $A_1, \ldots, A_t$ are independently random commitments of independently random values. □

4 Our Technical Contributions

4.1 An Improved Commitment Multiplication Proof

Consider again Pedersen's commitment scheme $\mathsf{com}_{g,h}(s,r) = g^s h^r$, and let $C' \in G$ be an arbitrary commitment. Then the commitment scheme

$$\mathsf{com}_{C',h}(s^*,r^*) = (C')^{s^*} h^{r^*} = C'^{s^*} \cdot \mathsf{com}_{g,h}(0,r^*)$$

with basis C', h inherits the following properties.

Lemma 1.

1. *Being able to open (wrt.* $\mathsf{com}_{g,h}$*) C' and C'' to values s' and s'', respectively, allows to open C'' wrt.* $\mathsf{com}_{C',h}$ *to a value s satisfying $ss' = s''$, and being able to open C'' to 0 wrt.* $\mathsf{com}_{g,h}$ *allows to open C'' to 0 wrt.* $\mathsf{com}_{C',h}$.
2. *The scheme* $\mathsf{com}_{C',h}$ *is as hiding and binding as* $\mathsf{com}_{g,h}$, assumed that C' cannot be opened to 0 wrt. $\mathsf{com}_{g,h}$.

Proof. 1. Let s, s', s'', r', r'' satisfy $C' = \mathsf{com}_{g,h}(s',r')$, $C'' = \mathsf{com}_{g,h}(s'',r'')$ and $ss' = s''$. Then, for $r^* = r'' - sr'$ we have $\mathsf{com}_{g,h}(s'' - ss', r^*) = C'' \cdot C'^{-s}$ and hence $\mathsf{com}_{C',h}(s,r^*) = C'^s \cdot \mathsf{com}_{g,h}(0,r^*) = C''$. This also holds if $s = 0$ and thus $s'' = 0$, in which case $r^* = r''$.

2. First, for $r^* \in \mathbb{F}_q$ chosen at random, $\mathsf{com}_{C',h}(s^*,r^*)$ is clearly a random element of G, independent of s^*. Furthermore, knowing $s^* \neq \tilde{s}^*$ and r^* and $\tilde{r}^*$ such that $\mathsf{com}_{C',h}(s^*,r^*) = \mathsf{com}_{C',h}(\tilde{s}^*,\tilde{r}^*)$, i.e. $C'^{s^*} \cdot \mathsf{com}_{g,h}(0,r^*) = C'^{\tilde{s}^*} \cdot \mathsf{com}_{g,h}(0,\tilde{r}^*)$, allows to open the commitment C' to zero, namely $C' = \mathsf{com}_{g,h}(0,(r^* - \tilde{r}^*)/(\tilde{s}^* - s^*))$. □

This gives rise to the following CMP, which allows the prover to prove that he can open commitments C, C' and C'' to values s, s' and $s'' = ss'$, respectively. Note that the 4 steps can be executed in parallel.

Protocol $\mathsf{Mult\,Proof}_{g,h}(C, C'; C'')$

1. The prover executes $\mathsf{Proof}_{g,h}(C)$.
2. The prover executes $\mathsf{Proof}_{C',h}(C'')$ using the *same* sharing polynomial $f_s(X)$ as in the above step (but new independent commitments wrt. $\mathsf{com}_{C',h}$).
3. Every player verifies whether his shares from step 1. and 2. coincide and accuses the dealer if it does not hold. In case $t < n/2$ (but not $t < n/3$) the dealer responds by opening the two corresponding commitments in public.
4. The prover executes $\mathsf{Proof}_{g,h}(C'')$.

This protocol also appeared in [1]. However, the security proof given there did *not* cover the case where the prover can open C' to $s' = 0$, and therefore the protocol was extended to "also deal with the case $s' = 0$" by essentially adding another Pedersen VSS sharing. Our analysis shows that this is superfluous, and that the protocol as it stands is secure also in case $s' = 0$. Furthermore, we show that the case $s = 0$ is somewhat special. Namely, we show that if the prover can open the commitment C to $s = 0$, then he can execute the protocol even *without being able to open* C', as long as he can open C'' to $s'' = 0$. This of course also guarantees that $s'' = ss'$ (no matter what s' is), but, as we will see in the next section, it also opens the door for new constructions in this setting like proofs of partial knowledge.

Theorem 1. *The above protocol* $\mathsf{Mult\,Proof}_{g,h}(C, C'; C'')$ *is a perfectly sound zero-knowledge distributed verifier proof, non-interactive in case* $t < n/3$ *and non-interactive with accusing in case* $t < n/2$, *that the prover can open* C, C' *and* C'' *as values* s, s' *and* $s'' = ss'$, or *that he can open both* C *and* C'' *as* 0.

Proof. *Correctness:* Follows from point 1. of Lemma 1 and the correctness of the protocol $\mathsf{Proof}_{g,h}$.
Soundness: According to Proposition 1, from the information received during Step 1., the honest players can compute s and r with $C = \mathsf{com}_{g,h}(s, r)$. Also, from the information received during Step 2., the honest players can compute *the same* s and some r^* with $C'' = \mathsf{com}_{C',h}(s, r^*) = C'^{s} \cdot \mathsf{com}_{g,h}(0, r^*)$. Finally, from the information received during Step 3., the honest players can compute s'' and r'' with $C'' = \mathsf{com}_{g,h}(s'', r'')$. It now follows that either $s = 0$ and hence $C'' = \mathsf{com}_{g,h}(0, r^*)$, which means that the honest players can open C'' to zero, or that $C' = C''^{1/s} \cdot \mathsf{com}_{g,h}(0, r^*)^{-1/s} = \mathsf{com}_{g,h}(s'', r'')^{1/s} \cdot \mathsf{com}_{g,h}(0, r^*)^{-1/s} = \mathsf{com}_{g,h}(s''/s, (r'' - r^*)/s)$, which means that the honest players can open C' to $s' = s''/s$.
Zero-Knowledge: The adversary can simulate his view of the protocol by simulating independently the protocols $\mathsf{Proof}_{g,h}(C)$, $\mathsf{Proof}_{C',h}(C'')$ and $\mathsf{Proof}_{g,h}(C'')$, as described in the proof of Proposition 1, *except* that he chooses the same shares for the simulation of $\mathsf{Proof}_{g,h}(C)$ and of $\mathsf{Proof}_{C',h}(C'')$. $\square$

4.2 Proofs or Partial Knowledge

In [8], an efficient solution was presented to construct proofs of *partial knowledge* in the two-players setting. Such a proof of partial knowledge allows for instance to prove the knowledge of (at least) ℓ out of m given secrets without revealing which ℓ secrets. We will now present corresponding non-interactive protocols in the distributed-verifier setting. While the proof protocols of [8] rely on concepts like the dual access structure and the simulation of protocols, our distributed verifier proof protocols are based on the fact that the CMP protocol $\mathsf{Mult\,Proof}_{g,h}(C, C'; C'')$ can be executed by the prover even if he does not know s' (as long as $s = s'' = 0$).

Let first C_0 and C_1 be two public Pedersen commitments and let the prover be able to open C_w to say s_w, where either $w = 0$ or $w = 1$.

Protocol $\mathsf{OR\text{-}Proof}_{g,h}(C_0, C_1)$

The prover sets $b_w = 1$ and $b_{1-w} = 0$ as well as $d_w = s_w$ and $d_{1-w} = 0$, and he commits to b_0, b_1, d_0 and d_1 by B_0, B_1, D_0 and D_1, respectively. Then, he opens $B = B_0 \cdot B_1$ as $b_0 + b_1 = 1$ and executes $\mathsf{Mult\,Proof}_{g,h}(B_0, C_0; D_0)$ and $\mathsf{Mult\,Proof}_{g,h}(B_1, C_1; D_1)$.

According to Theorem 1, the prover can execute $\mathsf{Mult\,Proof}_{g,h}(B_{1-w}, C_{1-w}; D_{1-w})$ even without being able to open C_{1-w} as long as he can open B_{1-w} and D_{1-w} to zero. On the other hand, if he cannot open B_w to zero, which must be the case for at least one of B_0 and B_1 as he can open $B = B_0 \cdot B_1$ to 1, $\mathsf{Mult\,Proof}_{g,h}(B_w, C_w; D_w)$ proves that he can open C_w.

This can easily be generalized to ℓ-out-of-m proofs, which, given m commitments $C_1, \ldots, C_m$, allows to prove the knowledge of at least ℓ hidden secrets, without giving away which ones.

Protocol $\binom{\ell}{m}$-$\mathsf{Proof}_{g,h}(C_1, \ldots, C_m)$

> For $i = 1, \ldots, m$, the prover sets $b_i = 1$ and $d_i = s_i$ if he can open C_i (to s_i) and $b_i = d_i = 0$ otherwise, and he commits to b_i and d_i by B_i and D_i, respectively. He proves that indeed $b_i \in \{0, 1\}$, i.e. $b_i(1 - b_i) = 0$, by executing $\mathsf{Mult\,Proof}_{g,h}(B_i, E/B_i; O)$, where $E = \mathsf{com}_{g,h}(1, 0) = g$ and $O = \mathsf{com}_{g,h}(0, 0) = 1$ are default commitments for 1 and zero, respectively, he opens $B_1 \cdot \ldots \cdot B_m$ as ℓ and executes $\mathsf{Mult\,Proof}_{g,h}(B_i, C_i; D_i)$ for $i = 1, \ldots, m$.

The following is a somewhat more efficient solution where no proof of something like $b_i \in \{0, 1\}$ is needed. Consider Shamir's ℓ-out-of m secret sharing scheme. As we have already used in the proof of Proposition 1, for $A \subseteq \{1, \ldots, m\}$ with $|A| \geq \ell$, there exist reconstruction coefficients $\lambda_{A,i}$, $i \in A$, such that $\sum_{i \in A}(\lambda_{A,i} i^k) = \delta_{k0}$. Based on this fact, we have the following enhanced protocol that allows the prover to prove that he can open the commitments C_i with $i \in A$ for a subset $A \subseteq \{1, \ldots, m\}$ of size at least ℓ.

Protocol $\binom{\ell}{m}$-$\mathsf{Proof}'_{g,h}(C_1, \ldots, C_m)$

> The prover chooses reconstruction coefficients $\lambda_{A,i}$, $i \in A$. For $i = 1, \ldots, m$, he puts $b_i = \lambda_{A,i}$ and $d_i = b_i s_i$ if $i \in A$ and $b_i = d_i = 0$ otherwise, and he generates commitments $B_1, \ldots, B_m$ and $D_1, \ldots, D_m$ for $b_1, \ldots, b_m$ and $d_1, \ldots, d_m$, respectively. For $k = 0, \ldots, \ell$, he opens the commitment $\prod_{i=1}^{m} B_i^{i^k}$ as δ_{k0}, and he executes $\mathsf{Mult\,Proof}_{g,h}(B_i, C_i; D_i)$ for $i = 1, \ldots, m$.

Soundness of the above protocol relies on the binding property of the Pedersen commitment scheme (hence it allows small error probability).

It is not hard to see that this protocol can be generalized to any linear secret sharing scheme, not necessarily a threshold scheme. Hence, given an arbitrary linear secret sharing scheme over $\mathbb{F}_q$ for m players with an access structure Γ, we have the following

Theorem 2. *Under the DL-assumption, there exists a computationally sound zero-knowledge distributed-verifier proof, non-interactive in case $t < n/3$ and non-interactive with accusing in case $t < n/2$, that the prover can open a subset $C_{i_1}, \ldots, C_{i_\ell}$ of the commitments $C_1, \ldots, C_m$ corresponding to a qualified set $A = \{i_1, \ldots, i_\ell\} \in \Gamma$.*

4.3 General Circuit Evaluation Proofs

Let $\mathcal{C}$ be a binary circuit consisting of NAND gates.

Theorem 3. *Under the DL-assumption, there exists a computationally sound zero-knowledge distributed-verifier proof, non-interactive in case $t < n/3$ and non-interactive with accusing in case $t < n/2$, that the prover knows a satisfying input to the circuit $\mathcal{C}$.*

Proof sketch: Let $b = (b_1, \dots, b_m)$ be a satisfying input for the circuit $\mathcal{C}$. To prove knowledge of b, the prover generates a commitment B_i for every input bit b_i and proves that $b_i \in \{0, 1\}$ by executing $\mathsf{Mult\,Proof}_{g,h}(B_i, E/B_i; O)$. Inductively, for every NAND gate with input bits b_l and b_r to which he has already computed corresponding commitments B_l and B_r, respectively, the prover computes a commitment B_{out} for the output bit $b_{out} = b_l$ NAND b_r and proves its correctness by executing $\mathsf{Mult\,Proof}_{g,h}(B_l, B_r; E/B_{out})$. Finally, he opens the commitment B of the result bit $b = \mathcal{C}(b_1, \dots, b_m)$ as 1. $\square$

Another way to achieve this result is by combining the techniques from [6] based on proofs of partial knowledge with the protocols from the above section.

Clearly, if the circuit $\mathcal{C}$ is an arithmetic circuit over the field $\mathbb{F}_q$, then there exists an even simpler proof protocol.

5 Arbitrary Homomorphic Commitments

In this section, we investigate to what extent the Pedersen's VSS scheme and the above results can be generalized with regard to other homomorphic commitment schemes. Clearly, by the description in Section 2.2, the Pedersen's VSS scheme, consisting of the protocols Share and $\mathsf{Reconstruct}$, can be executed with an arbitrary homomorphic commitment scheme replacing the Pedersen scheme. However, it is not so clear whether this results in a *secure* VSS scheme. And indeed, we will show that the security cannot be proven for an arbitrary (black-box) homomorphic commitment scheme. This does not necessarily imply that there exists a secure commitment scheme under which the Pedersen-like VSS is insecure; however, it means that in order to result in a secure Pedersen-like VSS, a homomorphic commitment scheme must inherit some additional properties. On the other hand, to relax the impact of this negative result, we present sufficient conditions for a homomorphic commitment scheme that guarantee the security of the corresponding Pedersen-like VSS and the resulting distributed-verifier proofs. We then show that these conditions are satisfied by so called q-one-way-group-homomorphism based schemes [7], which cover all currently known homomorphic commitment schemes with finite domain. Finally, we show how this positive result leads to error-free non-interactive zero-knowedge proofs of membership.

5.1 The Impossibility Result

Recall that a commitment scheme over a field K is called homomorphic if, given two commitments C and C' and a field element $\lambda \in K$, one can compute commitments S and P such that being able to open C and C' to values s and s', respectively, allows to open S to $s+s'$ and P to λs. We will denote these mappings $(C, C') \mapsto S$ and $(\lambda, C) \mapsto P$ by "$\star$" and "$\circ$", respectively, i.e. we write $S = C \star C'$ and $P = \lambda \circ C$. The following theorem states that the Pedersen VSS scheme described in Section 2.2 cannot be generalised to a homomorphic commitment scheme com, that is given as a *black-box* and where only the security requirements and the homomorphic property are guaranteed. The idea is that with respect to some unconditionally-hiding homomorphic commitment scheme, the dealer might be able to come up with commitments $A_0 = C, A_1, \ldots, A_t$ for the secret and the sharing coefficients, computed in some way such that he is not able to open (all of) them, but nevertheless he can open the corresponding commitments $C_1, \ldots, C_n$ to a set $s_1, \ldots, s_n$ of inconsistent shares. This is for instance the case if the dealer can compute a commitment A_1 such that he can open $2 \circ A_1, \ldots, (n-1) \circ A_1$ to $2, \ldots, n-1$, respectively, such that it looks as if A_1 "contains" 1, and $n \circ A_1$ to, let's say, $n+1$. Indeed, by choosing A_1 this way and $A_0 = C$ and $A_2, \ldots, A_t$ as required by the Share protocol, the dealer could open the corresponding commitments $C_2, \ldots, C_n$, computed as $C_i = C \star (i \circ A_1) \star \ldots \star (i^t \circ A_t)$, to a set of inconsistent shares (though he cannot open C_1). Since we do not require the dealer to be able to open A_1, and the homomorphic property does not require anything like $\lambda^{-1} \circ (\lambda \circ C) = C$ (as can be observed for existing schemes, see Section 5.2), the existence of such a commitment A_1 does not *a priori* contradict the security of the commitment scheme, if it is unconditionally hiding and hence a statement like "A_1 contains 1" does not make sense. We will now show that also *a posteriori*, this does not contradict the security (or the homomorphic property) of the commitment scheme by presenting an oracle with respect to which there exists a secure homomorphic commitment scheme, however the corresponding Pedersen-like VSS is insecure.

Theorem 4. *Let K be a field of size 2^k, where k is a security parameter. There exists an oracle $\mathcal{O}$ relative to which there exists a secure homomorphic commitment scheme $\mathsf{com}_{\mathcal{O}} : K \times K \to K$ such that the resulting Pedersen-like VSS, consisting of $\mathsf{Share}_{\mathcal{O}}$ and $\mathsf{Reconstruct}_{\mathcal{O}}$, is insecure.*

The oracle $\mathcal{O}$ in mind has history tapes $\mathcal{H}$, $\mathcal{M}$ and $\mathcal{A}$, which are all empty at the beginning, and one can make *commit-*, *multiply-*, *add-* and *cheat-queries*, to which $\mathcal{O}$ answers as follows:

commit-query: input $s, r \in K$, output $C = \mathsf{com}_{\mathcal{O}}(s, r) \in K$
If there exists $C \in K$ such that $(s, r; C) \in \mathcal{H}$, then $\mathcal{O}$ returns C. Else, $\mathcal{O}$ chooses a random $C \in K$, writes $(s, r; C)$ to the history tape $\mathcal{H}$ and returns C.

multiply-query: input $\lambda, C \in K$, output $C' = \mathsf{multiply}_{\mathcal{O}}(\lambda, C) \in K$
If there exists $C' \in K$ such that $(\lambda, C; C') \in \mathcal{M}$, then $\mathcal{O}$ returns C'. Else, if there exists $s, r \in K$ such that $(s, r; C) \in \mathcal{H}$, then $\mathcal{O}$ computes $C' =$

$\mathsf{com}_{\mathcal{O}}(\lambda s, \lambda r)$, while otherwise it chooses $C' \in K$ at random, and it writes $(\lambda, C; C')$ to the history tape $\mathcal{M}$ and returns C'.

add-query: input $C, C' \in K$, output $C'' = \mathsf{add}_{\mathcal{O}}(C, C') \in K$

If there exists $C'' \in K$ such that $(C, C'; C'') \in \mathcal{A}$, then $\mathcal{O}$ returns C''. Else, if there exist $s, r, s', r' \in K$ such that $(s, r; C), (s', r'; C) \in \mathcal{H}$, then $\mathcal{O}$ computes $C'' = \mathsf{com}_{\mathcal{O}}(s+s', r+r')$, while otherwise it chooses $C' \in K$ at random, and it writes $(C, C'; C'')$ to the history tape $\mathcal{A}$ and returns C''.

cheat-query: input $n \in \mathbb{N}$, output $(r^{(2)}, \ldots, r^{(n)}; C, C^{(2)}, \ldots, C^{(n)}) \in K^{n-1} \times K^n$

$\mathcal{O}$ chooses random $r^{(2)}, \ldots, r^{(n)} \in K$ and $C, C^{(2)}, \ldots, C^{(n)} \in K$. For $i = 2$ to n, he writes $(i, C; C^{(i)})$ to the history tape $\mathcal{M}$. For $i = 2$ to $n-1$, he writes $(i, r^{(i)}; C^{(i)})$ to the history tape $\mathcal{H}$, and he writes $(n+1, r^{(n)}; C^{(n)})$ to the history tape $\mathcal{H}$. Finally, he returns $r^{(2)}, \ldots, r^{(n)}$ and $C, C^{(2)}, \ldots, C^{(n)}$.

This oracle gives indeed rise to a homomorphic commitment scheme $\mathsf{com}_{\mathcal{O}} : K \times K \to K$. Namely, as indicated by the notation, for $s, r \in K$, the commitment $\mathsf{com}_{\mathcal{O}}(s, r)$ is the answer of the oracle $\mathcal{O}$ to a commit-query with input s and r, and the multiply- and add-queries provide the homomorphic property. E.g. being able to open C to s, i.e. knowing r such that $(s, r; C) \in \mathcal{H}$, allows to open $\lambda \circ C = \mathsf{multiply}_{\mathcal{O}}(\lambda, C)$, the answer C' to a multiply-query with input λ and C, to the value λs, since after the query $(\lambda s, \lambda r; C') \in \mathcal{H}$ and hence $\mathsf{com}_{\mathcal{O}}(\lambda s, \lambda r) = C'$. Furthermore, the cheat-query allows the dealer (together with a corrupted first player P_1) to misbehave as described in the beginning of this section to distribute an inconsistent sharing among the remaining players $P_2, \ldots, P_n$. It remains to show the security of com_{O}. The commitment C of a secret s, generated with whatever query, is a random number in K, independent of anything else, and hence the scheme is hiding. Because of the same reason, $C \neq C'$ for every pair $(s, r, C), (s', r', C')$ of entries of $\mathcal{H}$, except with small probability, and hence the scheme is binding.

It is not hard to see from the above construction that with respect to this homomorphic commitment scheme $\mathsf{com}_{\mathcal{O}}$, Proposition 1 and similarly Theorem 1 to 3 do not hold.

5.2 Generalization to q-OWGH-based Commitments

Inspecting for instance the proof of Proposition 1, which is essentially identical to a security proof of Pedersen's VSS scheme, one immediately sees that we made extensive use of the fact that for Pedersen's commitment scheme the operation "$\star$" is a group operation "$\cdot$", and that "$\circ$", given by exponentiation, fulfils

$$(C \cdot C')^{\lambda} = C^{\lambda} \cdot C'^{\lambda}, \quad C^{\lambda+\lambda'} = C^{\lambda} \cdot C^{\lambda'} \quad \text{and} \quad C^{\lambda\lambda'} = \left(C^{\lambda}\right)^{\lambda'}$$

which may not hold for other homomorphic schemes. In fact, with respect to the schemes listed in the appendix, this holds *only* for Pedersen's. For instance, if C is a commitment with respect to the QR-based commitment scheme $\mathsf{com}_t(s, r) = t^s r^2$ over $\mathbb{F}_2 = \{\bar{0}, \bar{1}\}$ ($\bar{x}$ denotes the residue class of x modulo q hereafter), then in general $C^{\bar{1}} \cdot C^{\bar{1}} = C \cdot C = C^2 \neq 1 = C^{\bar{0}} = C^{\bar{1}+\bar{1}}$. On the other hand, it is not hard to see that these were the *sole* conditions needed (besides the homomorphic property), not only for the proof of Proposition 1, but also for all results from the Sections 4. Hence, the above properties give a sufficient condition for a homomorphic commitment scheme in order to generalize the security of Pedersen's VSS scheme as well as our results to this commitment scheme. And, as a matter of fact, even some weaker condition suffices (which, by the way, are fulfilled by the above QR-based scheme, as will be shown):

It should be feasible to open the commitments

$$(C \cdot C')^{\lambda} \,/\, (C^{\lambda} {\cdot} C'^{\lambda}) \,, \quad C^{\lambda+\lambda'} \,/\, (C^{\lambda} {\cdot} C^{\lambda'}) \quad \textit{and} \quad C^{\lambda\lambda'} \,/\, \big(C^{\lambda}\big)^{\lambda'} \tag{3}$$

to zero for any commitments C, C' and numbers λ, λ', knowing only C, C', λ and λ'.

Indeed, consider for instance (1) in the proof of Proposition 1. Even though it might be that $C' \neq C$, it is guaranteed by these properties that the commitment C/C' can be opened to zero knowing only C and C', and hence being able to open C' (to s') also allows to open $C = (C/C')\cdot C'$ (to s'). This kind of reasoning allows to generalize all the previous proofs, and hence we have

Proposition 2. *The security of Pedersen's VSS scheme as well as Proposition 1 and Theorem 1 to 3 hold for every homomorphic commitment scheme satisfying the above condition (3).*

We will now show that all q-one-way-group-homomorphism based commitment schemes, which contain all so far known homomorphic schemes with finite domain, fulfil this condition (3). We start by recalling the concept of q-one-way-group-homomorphism. Let q be a prime number. Loosely speaking, a *q-one-way-group-homomorphism*, q-OWGH for short, is a homomorphism $f : H \to G$ among two finite Abelian groups H and G, such that f is one-way, but, for a randomly chosen $y \in G$, it is feasible to compute $v \in H$ with $f(v) = y^q$. For formal definitions we refer to [7], where this concept was introduced.

Such a q-OWGH induces in a generic way a computationally binding commitment scheme over the field $\mathbb{F}_q$. Namely the scheme

$$\mathsf{com}_{g,f} : \mathbb{F}_q \times H \to G,\ (s,r) \mapsto g^s f(r)$$

where g is randomly chosen from $\mathrm{im}(f) \subseteq G$ and g^s is defined as g^{ς} with $\varsigma \in \{0,\dots,q-1\}$ such that $\overline{\varsigma} = s$. Note, it is *not* required that G has order q.

If a q-OWGH $f : H \to G$ is *unconditionally binding* [7], meaning that there exists $t \in G$ such that t has order q modulo $im(f)$ and $t^i f(r)$ and $t^j f(s)$ are computationally indistinguishable for all i and j and for randomly (and independently) chosen r and s, then f also induces a computationally hiding commitment scheme over $\mathbb{F}_q$. Namely,

$$\mathsf{com}_{t,f} : \mathbb{F}_q \times H \to G,\ (s,r) \mapsto t^s f(r)$$

for such a particular $t \in G$.

For the security proof of these commitments, we refer to [7].

An important property of these commitment schemes is that they are homomorphic. Indeed, if $C = \mathsf{com}(s, r) = g^s f(r)$ and $C' = \mathsf{com}(s', r) = g^{s'} f(r')$ and $\lambda \in \mathbb{F}_q$, we have, writing $s = \overline{\varsigma}$, $s' = \overline{\varsigma'}$, $\lambda s + s' = \overline{\varsigma''}$ and $\lambda = \overline{\ell}$ with $\varsigma, \varsigma', \varsigma'', \ell \in \{0, \ldots, q-1\}$ as well as $\ell\varsigma + \varsigma' = kq + \varsigma'' \in \mathbb{Z}$,

$$C^\lambda C' = (g^\varsigma f(r))^\ell g^{\varsigma'} f(r') = g^{\ell\varsigma+\varsigma'} f(\ell r + r') = g^{kq+\varsigma''} f(\ell r + r') = g^{\lambda s + s'} f(kv + \ell r + r')$$

where $v \in H$ is computed such that $f(v) = g^q$.

Lemma 2. *For any q-OWGH based commitment scheme, any commitments C and C' and numbers $\lambda, \lambda' \in \mathbb{F}_q$, the commitments*

$$(C \cdot C')^\lambda \,/\, (C^\lambda \cdot C'^\lambda)\,, \quad C^{\lambda+\lambda'} \,/\, (C^\lambda \cdot C^{\lambda'}) \quad \text{and} \quad C^{\lambda\lambda'} \,/\, \left(C^\lambda\right)^{\lambda'}$$

can be opened to zero knowing only C, C', λ and λ'.

Proof. Clearly, $(C \cdot C')^\lambda = (C^\lambda \cdot C'^\lambda)$ and thus $(C \cdot C')^\lambda / (C^\lambda \cdot C'^\lambda) = 1 = \mathsf{com}_{g,f}(0, 0)$. Furthermore, if $\lambda = \overline{\ell}$, $\lambda' = \overline{\ell'}$ and $\lambda\lambda' = \overline{\ell''}$ with $\ell, \ell', \ell'' \in \{0, \ldots, q-1\}$ and $\ell\ell' = kq + \ell''$, we get

$$C^{\lambda\lambda'} / (C^\lambda)^{\lambda'} = C^{\ell' - \ell\ell'} = C^{-kq} = f(-kv) = \mathsf{com}_{g,f}(0, -kv)$$

where $v \in H$ is computed such that $f(v) = C^q$. And of course, the same argument can be applied to $C^{\lambda+\lambda'} / C^\lambda C^{\lambda'}$. $\square$

It now follows from Proposition 2

Theorem 5. *The security of Pedersen's VSS scheme as well as Proposition 1 and Theorems 1 to 3 hold for every q-OWGH based commitment scheme.*

Note that Shamir's secret sharing scheme does not work over $\mathbb{F}_q$ if $q \le n$. Hence, in this case, Pedersen's VSS and the resulting proof protocols have to be based on a different linear secret sharing scheme. However, it is straight forward to verify that replacing Shamir's secret sharing scheme in Pedersen's VSS and the resulting proof protocols by an arbitrary linear secret sharing scheme [14] does not affect any of the results. This also allows to generalize the results to arbitrary (not necessarily threshold) adversary structures [13].

5.3 On Proofs of Membership

In this last section, we show that in the distributed-verifier setting there exist error-free non-interactive zero-knowledge proofs of membership for non-trivial languages, which is well known not to exist in the usual single-verifier setting.

Recall the protocol Proof from Section 3.2, but now based on an arbitrary q-OWGH based commitment scheme $\mathsf{com} : \mathbb{F}_q \times H \to G$. It allows the dealer to prove that he can open a given commitment C to *some* value. Assume now that he wants to prove that he can open C to a concrete given value s, e.g. $s = 0$.

This can be done simply by executing the protocol Proof as given, except that the dealer uses the default sharing polynomial $f_s(X) = s$ (instead of a random one), such that every share coincides with s (and if this is not the case for some player then he accuses the dealer). We denote this modified protocol by Proof'. It can be shown similarly to the proof of Proposition 1 that this indeed proves that the dealer can open C to s, or, in terms of proofs of membership, that C is a commitment of s:

Proposition 3. *Protocol* Proof' *is a perfectly sound zero-knowledge distributed-verifier proof that* C *is in* $\{\mathsf{com}(s, r) \mid r \in H\} \subseteq G$*, non-interactive in case* $t < n/3$ *and non-interactive with accusing in case* $t < n/2$.

Using unconditionally binding commitment schemes like the QR- or the DCR-based ones described in the appendix, this results in error-free non-interactive proofs for non-trivial subgroup membership problems: The former allows to prove that a given number is a quadratic residue modulo an RSA modulus n, and the latter that a given number is an n-th power modulo n^2, simply by proving that the number is a commitment of $s = 0$.

In fact, one can construct proves for arbitrary subgroup membership problems (even if they do not result from homomorphic commitments), i.e., proves that allow to prove that a group element $C \in G$ belongs to a subgroup $G' \subset G$, as long as for every $C \in G'$ there exists a corresponding witness w in a group H such that the mapping $\varphi : H \to G'$, $w \mapsto C$ is a group homomorphism. Namely, by executing Proof using $\mathsf{com} = \varphi : H \times \emptyset \to G'$ as "commitment": The dealer chooses random witnesses $a_1, \ldots, a_t \in H$, publishes the corresponding subgroup elements $A_k = \varphi(a_k) \in G'$, $k = 1, \ldots, t$, and sends the witness for $C_i = C \cdot A_1^i \cdot \ldots \cdot A_t^{i^t} \in G'$ privately to player P_i, $i = 1, \ldots, n$. For instance, this way one can prove that a triple $(u, v, w) \in G^3$ is a Diffie-Hellman triple with respect to g, i.e. that $(u, w) \in \{(g^a, v^a) \,|\, a \in \mathbb{F}_q\} \subset G \times G$. Note, in this example, $\varphi : \mathbb{F}_q \to G^2$, $a \mapsto (g^a, v^a)$.
If the order of the group H is not known, then Shamir's secret sharing scheme can be replaced by a black box secret sharing scheme [9].

References

1. Masayuki Abe. Robust distributed multiplication without interaction. In *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.
2. Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2), 1988.
3. Donald Beaver. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2), 1991.
4. Mihir Bellare, and Oded Goldreich. On Defining Proofs of Knowledge. In *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*. Springer, 1998.
5. Joan F. Boyar, Mark W. Krentel, and Stuart A. Kurtz. A discrete logarithm implementation of zero-knowledge blobs. Technical Report TR-87-02, Department of Computer Science, University of Chicago, 1987.

6. Ronald Cramer and Ivan Damgård. L Linear zero-knowledge: A note on efficient zero-knowledge proofs and arguments. In *29th ACM Symposium on Theory of Computing*. ACM Press, 1997.
7. Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic or: Can zero-knowledge be for free? In *Advances in Cryptology—CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*. Springer, 1998.
8. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology—CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*. Springer, 1994.
9. Ronald Cramer and Serge Fehr. Optimal black-box secret sharing over arbitrary Abelian groups. In *Advances in Cryptology—CRYPTO '02*, volume 2442 of *Lecture Notes in Computer Science*. Springer, 2002.
10. Ronald Cramer, Matthew Franklin, Berry Schoenmakes, and Moti Yung. Multi-authority secret-ballot elections with linear work. In *Advances in Cryptology—EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*. Springer, 1996.
11. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*. IEEE, 1985.
12. Rosario Gennaro, Michael O. Rabin, and Tal Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *17th ACM Symposium on Principles of Distributed Computing*, 1998.
13. Martin Hirt and Ueli Maurer Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In *16th ACM Symposium on Principles of Distributed Computing*, 1997. Final version appeared in Journal of Cryptology 2000.
14. Maurizio Karchmer and Avi Wigderson. On span programs. In *8th Annual Conference on Structure in Complexity Theory*. IEEE, 1993.
15. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology—EUROCRYPT '99*, Lecture Notes in Computer Science. Springer, 1999.
16. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. Springer, 1991.
17. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11), 1979.

A Examples of q-OWGH Based Commitments

Based on the DL-problem:

Let p be prime and $G = \langle h \rangle$ a subgroup of $\mathbb{Z}_p^*$ with prime order q. Then the exponentiation function $f : \mathbb{Z}_{q-1} \to G$, $x \mapsto h^x$ is (a candidate for) a q-OWGH. Indeed, given $y \in G$, $v = 0$ fulfils $f(v) = 1 = y^q$.

The resulting commitment scheme is the Pedersen commitment scheme we were considering in the first part.

Based on the RSA-problem:
The RSA function $f : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$, $x \mapsto x^q$ for a prime exponent q is (a candidate for) a q-OWGH. Given $y \in \mathbb{Z}_n^*$, $v = y$ fulfils $f(v) = y^q$.
The resulting commitment scheme is $\mathsf{com}_{g,f}(s,r) = g^s r^q$.

Based on factoring and on the QR-problem:
Squaring modulo an RSA modulus n, $f : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$, $x \mapsto x^2$ is (a candidate for) an unconditionally binding 2-OWGH. Given $y \in \mathbb{Z}_n^*$, $v = y$ fulfils $f(v) = y^2$ and any quadratic non-residue $t \in \mathbb{Z}_n^*$ with Jacoby symbol $+1$ fulfils the requirements for the unconditionally binding property, assumed that the QR-problem is hard. The resulting computationally binding commitment scheme is $\mathsf{com}_{g,f}(s,r) = g^s r^2$ for a random quadratic residue g and the resulting computationally hiding scheme is $\mathsf{com}_{t,f}(s,r) = t^s r^2$ for a quadratic non-residue t with Jacoby symbol $+1$, both occurring in [2].

Based on computing n-th roots mod n^2 and on the DCR assumption:
The function $f : \mathbb{Z}_{n^2}^* \to \mathbb{Z}_{n^2}^*$, $x \mapsto x^n$ for an RSA modulus n is (a candidate for) an unconditionally binding n-OWGH. Given $y \in \mathbb{Z}_{n^2}^*$, $v = y$ fulfils $f(v) = y^n$ and e.g. $t = \overline{n+1} \in \mathbb{Z}_{n^2}^*$ fulfils the requirements for the unconditionally binding property, based on the decisional composite residuosity (DCR) assumption [15]. The resulting computationally binding commitment scheme is $\mathsf{com}_{g,f}(s,r) = g^s r^n$ for a random n-th power g and the resulting computationally hiding scheme is $\mathsf{com}_{t,f}(s,r) = t^s r^n$ for e.g. $t = \overline{n+1} \in \mathbb{Z}_{n^2}^*$, i.e. the Paillier encryption function [15].
Note that even though n is not a prime, it can be treated in this context as one, as it is (assumed to be) hard to find non-trivial divisors.