Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing

Torben Pryds Pedersen
Computer Science Department
Aarhus University, Denmark
tppedersen@daimi.aau.dk

Abstract

It is shown how to distribute a secret to n persons such that each person can verify that he has received correct information about the secret without talking with other persons. Any k of these persons can later find the secret $(1 \leq k \leq n)$, whereas fewer than k persons get no (Shannon) information about the secret. The information rate of the scheme is $\frac{1}{2}$ and the distribution as well as the verification requires approximately $2k$ modular multiplications pr. bit of the secret. It is also shown how a number of persons can choose a secret "in the well" and distribute it verifiably among themselves.

1 Introduction

Secret sharing schemes were introduced independently in [Sha79] and [Bla79] and since then much work has been put into the investigation of such schemes (see [Sim90] for a list of references). The verifiable secret sharing schemes constitute a particular interesting class of these schemes as they allow each receiver of information about the secret (*share* of the secret) to verify that the share is consistent with the other shares.

Let the *dealer* be the person who has a secret and distributes it to n *shareholders*, where $n > 0$. If the dealer trusts one of the shareholders completely, he could give the secret to this person and then avoid the troubles of having a secret sharing scheme. Thus in many applications the dealer does not trust the shareholders completely, and therefore it should be expected that (some of) the shareholders do not trust the dealer either. For this reason efficient verifiable secret sharing schemes are necessary in practice.

However, verifiable secret sharing has also turned out to be a useful tool in more theoretical work. In [BGW88] and [CCD88] unconditionally secure verifiable secret sharing schemes are constructed and used to design secure multi-party protocols. Unfortunately, these schemes are *interactive* — interaction between the participants is needed in order to verify the shares. Both of these schemes require that less than $\frac{n}{3}$ of the shareholders are dishonest. This is improved in [RB89], where a scheme with the same properties is presented, except that it allows less than $\frac{n}{2}$ dishonest participants. These three schemes

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 129-140, 1992.

all have the property that even an all powerful dealer cannot distribute incorrect shares (in [CCD88] and [RB89] there is an exponentially small error probability however).

In this paper, we are mainly interested in *non-interactive* verifiable secret sharing. In such a scheme only the dealer is allowed to send messages — in particular the shareholders cannot talk with each other or the dealer when verifying a share. This model is very suitable in practice as it allows distribution by mail for instance.

[Ben87] presented the first non-interactive verifiable secret sharing scheme, but it relied on the existence of a mutually trusted entity. In [Fel87] this entity is avoided by letting the dealer publish probabilistic encryptions of the polynomial used to compute the shares, and due to a homomorphism property of the encryption scheme verification of the shares is possible. This scheme is quite efficient, but after the distribution, the privacy of the secret depends on a computational assumption — such as the intractability of computing discrete logarithms.

The goal of this paper is to construct an efficient non-interactive scheme for verifiable secret sharing in which no (Shannon) information about the secret is revealed. [Ped91] presents a non-interactive verifiable secret sharing scheme which can be used for secrets, s, for which g^s is known, where g is the generator of a group. In this paper the scheme suggested in [Ped91] is modified in order to remove the assumption that g^s is known beforehand. This results in a secret sharing scheme which is unconditionally secure for the dealer. However, in this scheme the dealer can succeed in distributing incorrect shares, if he can solve the discrete logarithm problem (see [BM84] for a formal definition). This property is inevitable as we shall see that it is impossible to construct a non-interactive secret sharing scheme in which no information about the secret is revealed and even a dealer with unlimited computing power cannot cheat. Thus this scheme is in some sense dual to that of [Fel87] (see Section 4.3).

The new secret sharing scheme is constructed by combining Shamir's scheme (see [Sha79]) with a commitment scheme, which is unconditionally secure for the committer and furthermore allows commitment to many bits simultaneously. This commitment scheme is a variant of a scheme proposed in [BCP].

After introducing some notation in Section 2, Section 3 describes the commitment scheme, and in Section 4 the secret sharing scheme is presented. As an application of this scheme, Section 5 shows how the shareholders can compute linear combinations of shared secrets and Section 6 concludes the paper.

2 Notation

Throughout this paper p and q denote large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q, and g is a generator of G_q. It can easily be tested if an element $a \in \mathbb{Z}_p^*$ is in G_q since

$$a \in G_q \iff a^q = 1.$$

As any element $b \neq 1$ in G_q generates the group, the discrete logarithm of $a \in G_q$ with respect to the base b is defined and it is denoted $\log_b(a)$.

For any integer x the length of the binary representation of x is denoted $|x|$.

3 The Commitment Scheme

This section describes a commitment scheme, which is very similar to that of [BCP]. The only difference is in the choice of g and h.

Let g and h be elements of G_q such that nobody knows $\log_g h$. These elements can either be chosen by a trusted center, when the system is initialized, or by (some of) the participants using a coin-flipping protocol.

The committer commits himself to an $s \in \mathbb{Z}_q$ by choosing $t \in \mathbb{Z}_q$ at random and computing

$$E(s,t) = g^s h^t.$$

Such a commitment can later be opened by revealing s and t. The following theorem is very easy to prove and shows that $E(s,t)$ reveals no information about s, and that the committer cannot open a commitment to s as $s' \neq s$ unless he can find $\log_g(h)$.

Theorem 3.1
For any $s \in \mathbb{Z}_q$ and for randomly uniformly chosen $t \in \mathbb{Z}_q$, $E(s,t)$ is uniformly distributed in G_q.
If $s, s' \in \mathbb{Z}_q$ satisfies $s \neq s'$ and $E(s,t) = E(s',t')$, then $t \neq t' \bmod q$ and

$$\log_g h = \frac{s - s'}{t' - t} \bmod q.$$

Even though it will not be used in the following we mention that it is quite easy to prove one's ability to open two commitments as the same value without revealing this value. Let namely

$$\beta = E(s,t) \quad \text{and} \quad \beta' = E(s,t')$$

where $t \neq t'$. Anyone who knows an r such that $\beta/\beta' = h^r$ can open β as s if and only if he can also open β' as s. By revealing $r = t - t'$ it is therefore possible to prove equality of the contents of two commitments. Furthermore, $t - t'$ does not contain any information about s. It is not clear how to prove efficiently, that commitments to two different values really do contain different values. In particular, the proof of [BCC88] that two blobs contain different bits given a method of proving equality does not generalize to this commitment scheme.

Finally consider the efficiency of the commitment scheme. If p and q are constructed by first choosing q and then determining p as the first prime congruent to $1 \bmod q$, heuristics show that $p \leq q(\log q)^2$ (see [Wag79]). Thus a commitment to $|q|$ bits requires at most $|q| + 2\log|q|$ bits. Furthermore, by first computing the product gh a commitment to s can be done in less than $2|q|$ multiplications modulo p or less than two multiplications pr. bit of s. Thus the commitment scheme is quite efficient with respect to the size of commitments as well as the computation required.

4 Non-interactive Verifiable Secret Sharing

This section first defines verifiable secret sharing, and then the commitment scheme described above and the Shamir scheme are combined resulting in a non-interactive verifiable secret sharing scheme. Finally, the efficiency of the scheme is estimated.

4.1 Verification of Shares

Assume that a dealer, D, has a secret $s \in \mathbb{Z}_q$ and wants to distribute it among n parties, $P_1, \ldots, P_n$, such that any k of the shareholders can find s if necessary, but less than k shareholders get no (Shannon) information about s (a (k,n)-threshold scheme). Shamir suggested that the dealer could do this by choosing a polynomial $f \in \mathbb{Z}_q[x]$ of degree at most $k-1$ such that $f(0) = s$ and then give P_i the share $f(i)$. $P_{i_1}, \ldots, P_{i_k}$ can later find s from the formula for f:

$$f(x) = \sum_{j=1}^{k} (\prod_{l \neq j} \frac{x - i_j}{i_l - i_j}) f(i_j)$$

as

$$s = \sum_{j=1}^{k} (\prod_{l \neq j} \frac{i_j}{i_j - i_l}) f(i_j).$$

Our goal is to extend this scheme with a verification protocol, VP, such that any k participants, who have (honestly) accepted their shares in VP can find s. More formally VP must satisfy:

Definition 4.1
A verification protocol, VP, takes place between the dealer and $P_1, \ldots, P_n$. It must satisfy the following two requirements:

1. If the dealer follows the distribution protocol and if the dealer and P_i both follow VP, then P_i accepts with probability 1.
2. For all subsets S_1 and S_2 of $\{1, \ldots, n\}$ of size k such that all parties $(P_i)_{i \in S_1}$ and $(P_i)_{i \in S_2}$ have accepted their shares in VP the following holds except with negligible probability in $|q|$: If s_i is the secret computed by the participants in S_i (for $i = 1, 2$) then $s_1 = s_2$.

A share is called *correct*, if it is accepted in VP.

Even though this definition allows any kind of interaction between the dealer and the participants we shall only be concerned with *non-interactive* verification protocols here. In this case the dealer sends extra information to each participant during the distribution, and in the verification protocol P_i verifies that his secret share is consistent with this extra information.

Definition 4.1 does not refer to the secret when defining the correctness of a share. This is in accordance with the fact that no participant have any information about s during the verification and therefore s could be whatever the dealer claims. After the execution of the verification protocol the secret is defined as the value, which any k participants with correct shares will find when combining their shares. If the dealer succeeds in distributing inconsistent shares, this is not well-defined, but Definition 4.1 guarantees that the dealer will be caught almost always when trying to cheat.

4.2 The Scheme

Let $g, h \in G_q$ be given such that the commitment scheme from Section 3 can be applied. By the fact that $\mathbb{Z}_q$ is a field, the dealer can distribute $s \in \mathbb{Z}_q$ as follows:

1. D publishes a commitment to s: $E_0 = E(s,t)$ for a randomly chosen $t \in \mathbb{Z}_q$.
2. D chooses $F \in \mathbb{Z}_q[x]$ of degree at most $k-1$ satisfying $F(0) = s$, and computes $s_i = F(i)$ for $i = 1, \ldots, n$.
   Let $F(x) = s + F_1 x + \ldots + F_{k-1}x^{k-1}$. D chooses $G_1, \ldots, G_{k-1} \in \mathbb{Z}_q$ at random and uses G_i when committing to to F_i for $i = 1, \ldots, k-1$. D broadcasts $E_i = E(F_i, G_i)$ for $i = 1, \ldots, k-1$.
3. Let $G(x) = t + G_1 x + \ldots + G_{k-1}x^{k-1}$ and let $t_i = G(i)$ for $i = 1, \ldots n$. Then D sends (s_i, t_i) secretly to P_i for $i = 1, 2, \ldots, n$.

When P_i has received his share (s_i, t_i) he verifies that

$$E(s_i, t_i) = \prod_{j=0}^{k-1} E_j^{i^j} \qquad (*).$$

Lemma 4.2
Let $S \subset \{1, \ldots, n\}$ be a set of k participants such that $(*)$ holds for these k parties. Then these k parties can find a pair (s', t') such that $E_0 = g^{s'}h^{t'}$.

Proof
Let $S \subseteq \{1, \ldots, n\}$ of size k be given. The participants in S first find the two unique polynomials F' and G' of degree at most $k-1$ satisfying

$$\begin{aligned} F'(i) &= s_i \\ G'(i) &= t_i \end{aligned}$$

for $i \in S$. Now let $h = g^d$. Then

$$g^{F'(i)+dG'(i)} = E(s_i, t_i) = g^{s_i + dt_i}$$

for $i \in S$. Thus $(F' + dG')(x)$ is the unique polynomial of degree at most $k-1$ mapping i to $s_i + dt_i$. Let $E_j = g^{e_j}$. Then the polynomial

$$e(x) = \sum_{j=0}^{k-1} e_j x^j$$

satisfies $e(i) = s_i + dt_i$ for $i \in S$. Thus

$$e(x) = (F' + dG')(x)$$

and in particular

$$E_0 = g^{e(0)} = g^{F'(0)+dG'(0)} = g^{F'(0)}h^{G'(0)}.$$

Therefore it is sufficient to put $s' = F'(0)$ and $t' = G'(0)$. ■

The members in S do not have to find F' in order to find the secret. It is more efficient to use the formula

$$s = \sum_{i \in S} a_i s_i \quad \text{where} \quad a_i = \prod_{j \in S, j \neq i} \frac{i}{i-j}.$$

Note that they can also find t by the formula

$$t = \sum_{i \in S} a_i t_i.$$

Theorem 4.3
Under the assumption that the dealer cannot find $\log_g h$ except with negligible probability in $|q|$, the verification protocol satisfies Definition 4.1.

Proof
It is not hard to see that $(*)$ will be satisfied for all participants if the dealer follows the protocol.

Let S and S' be two subsets of $\{1, \ldots, n\}$ of size k such that all participants in S and S' have accepted their shares correctly. According to Lemma 4.2 the members of S and S' can find pairs (s,t) and (s',t'), respectively, such that $E_0 = E(s,t) = E(s',t')$.

As the shares are consistent if and only if there is a polynomial, f, of degree at most $k-1$ such that

$$f(i) = s_i \qquad \text{for } i = 1, 2, \ldots, n$$

the dealer can find the two sets S and S' as follows, if the shares are inconsistent:

1. Let f be the unique polynomial of degree at most $k-1$ such that $f(i) = s_i$ for $i = 1, 2, \ldots, k$.
2. Let $i = k + 1$.
3. If $i > n$ then stop (all shares are consistent).
   If $f(i) = s_i$ then put $i := i + 1$ and goto 3.
   Otherwise return the sets $S = \{1, 2, \ldots, k\}$ and $S' = \{1, 2, \ldots, k-1, i\}$.

Thus, if the dealer has succeeded in distributing inconsistent shares, he can find $\log_g h$ by first finding S and S' as described above and then computing $\log_g h$ as in Theorem 3.1.
■

As a consequence of Theorem 4.3 all the shares satisfying $(*)$ are consistent unless the dealer succeeds in finding $\log_g(h)$ *before* the last share has been sent.

The following theorem shows, that fewer than k participants get no (Shannon) information about the secret. For any subset $S \subset \{1, \ldots, n\}$, $views_S$ denotes the messages, that the members of S see:

$$views_S = (E_0, E_1, \ldots, E_{k-1}, (s_i, t_i)_{i \in S}).$$

Theorem 4.4
For any $S \subset \{1, \ldots, n\}$ of size at most $k-1$ and any $views_S$

$$Prob[D \text{ has secret } s \mid views_S] = Prob[D \text{ has secret } s]$$

for all $s \in \mathbb{Z}_q$.

Proof
It is sufficient to prove the theorem in the case where S has size $k-1$. If $k-1$ parties do not get any information about s then neither does fewer than $k-1$ parties.

Let $S = \{1, \ldots, k-1\}$ and let $view_S = (E_0, E_1, \ldots, E_{k-1}, (s_i, t_i)_{i=1,\ldots,k-1})$. For every $s \in \mathbb{Z}_q$ there is exactly one $t \in \mathbb{Z}_q$ such that $E_0 = E(s,t)$ and there is exactly one polynomial F of degree at most $k-1$ satisfying

$$\begin{aligned} F(0) &= s \\ F(i) &= s_i \qquad \text{for } i = 1, \ldots, k-1 \end{aligned}$$

and exactly one polynomial G of degree at most $k-1$ satisfying

$$\begin{aligned} G(0) &= t \\ G(i) &= t_i \qquad \text{for } i = 1, \ldots, k-1 \end{aligned}$$

Let $F(x) = s + F_1 x + \ldots + F_{k-1} x^{k-1}$ and $G(x) = t + G_1 x + \ldots + G_{k-1} x^{k-1}$. In order to show that $view_S$ does not contain any information about the secret it must be shown that F and G satisfies

$$E(F_i, G_i) = E_i \qquad \text{for } i = 1, \ldots, k-1,$$

as this is true for the polynomials chosen by the dealer. As in the proof of Lemma 4.2 this follows from the fact that there is one and only one polynomial, f, of degree at most $k-1$ satisfying ($s_0 = s$, $t_0 = t$)

$$g^{f(i)} = g^{s_i} h^{t_i}$$

for $i = 0, 1, \ldots, k-1$ and the polynomial $F + dG$ satisfies this for $d = \log_g h$. ■

4.3 Efficiency and Security

In this section, the computational requirements of the scheme are estimated and the scheme is compared to [Fel87].

First consider the size of the secret shares. The information rate (see [BD90]) is

$$\frac{\text{size of secret}}{\text{size of share}} = \frac{1}{2}.$$

Ignoring the time needed to evaluate $F(x)$ and $G(x)$ (this is reasonable as the polynomials are only evaluated on small arguments), the dealer has to compute k commitments in order to verify a share. This requires less than $2|q|k$ multiplications modulo p or approximately $2k$ multiplications pr. bit of the secret, if every element in $\mathbb{Z}_q$ can be chosen as the secret.

The verification requires $k-1$ exponentiations modulo p and the computation of one commitment. This can be done in less than (again ignoring the computation of i^j for $j = 1, \ldots, k-1$)

$$2|q|(k-1) + 2|q| + (k-1) \approx (2|q|+1)k$$

multiplications. This is however, a pessimistic estimate as many of the exponents in the exponentiations are rather small (in particular, for P_1 they all equal 1).

The scheme presented here is in many respects similar to that of [Fel87], which works for any probabilistic encryption scheme in which a number of bits (say l) are encrypted as the "hard-core" bits of a one-way function with homomorphic properties. Specifically, it is suggested to use the function

$$x \mapsto g^x \qquad \text{for } x \in \mathbb{Z}_q^*$$

and encrypt $l = O(\log|q|)$ bits as g^x where the l bits in question are easy to compute from x. Using this scheme, the computational requirements when distributing an l-bits secret is very similar to the requirements in our scheme when distributing a $|q|$-bits secret (note that $|q| \approx 2^l$).

With respect to security the two schemes are dual to each other, because the encryption schemes used in [Fel87] only protects the secret under the assumption that the one-way function cannot be inverted. However, even an infinitely powerful dealer cannot distribute incorrect shares. In contrast, the new scheme protects the privacy of the secret unconditionally, but the correctness of the shares depends on a computational assumption.

Having these two secret sharing schemes it is natural to ask for a non-interactive scheme in which

- no information about the secret is revealed; and
- even an infinitely powerful dealer cannot compute inconsistent shares.

However, the following shows that such a scheme is impossible in the model which is used here. Let namely b denote all the information which the dealer broadcasts in a non-interactive secret sharing scheme, and let s_i be the secret share which is sent to P_i. Let $V(i, b, s_i)$ denote the verification predicate which P_i computes in order to verify his share. Now consider $P_1, \ldots, P_{k-1}$ and assume that they have received correct shares. Let S_k be the set of shares which P_k can receive:

$$S_k(b) = \{s_k \mid V(k, b, s_k)\}.$$

As even an all powerful dealer cannot find inconsistent shares then $P_1, \ldots, P_{k-1}, P_k$ will find the same secret for any $s_k \in S_k$. This means that $P_1, \ldots, P_{k-1}$ can find the secret by guessing a secret share $s_k \in S_k$ and then combine their own shares with s_k.

In particular note that $S_k(b)$ is in NP if V can be computed in polynomial time. Therefore does $P_1, \ldots, P_{k-1}$ "only" need nondeterministic polynomial time in order to find the secret if the scheme is unconditionally secure for the shareholders. Similarly, a dishonest dealer can distribute inconsistent shares in nondeterministic polynomial time if the scheme reveals no information about the secret.

5 Computing on Shared Secrets

As mentioned in the introduction verifiable secret sharing is an important tool in the construction of secure protocols for multiparty computations. In particular both the construction in [BGW88] and [CCD88] utilize the fact that it is easy to compute linear combinations of shared secrets. In this section we show that this is also true if the secret sharing scheme presented here is used, and we present an application of this property.

5.1 Linear Combinations

Assume that two secrets s' and s'' have been distributed as described in Chapter 4. In particular let (s_i', t_i') and (s_i'', t_i'') be P_i's share of s' and s'', respectively, and let $(E_0', E_1', \ldots, E_{k-1}')$ and $(E_0'', E_1'', \ldots, E_{k-1}'')$ be the broadcasted messages when the two secrets were distributed.

Each P_i can compute $(E_0, E_1, \ldots, E_{k-1})$ corresponding to a verifiable distribution of $s = s' + s'' \bmod q$ as

$$E_j = E_j' E_j'' \qquad \text{for } j = 0, 1 \ldots k-1.$$

Furthermore, P_i's secret share, (s_i, t_i), of s is given by

$$\begin{aligned} s_i &= s_i' + s_i'' \bmod q \\ t_i &= t_i' + t_i'' \bmod q \end{aligned}$$

By insertion it is easy to see that if both (s_i', t_i') and (s_i'', t_i'') are correct shares (satisfy $(*)$) then (s_i, t_i) is also a correct share of s; i.e.

$$g^{s_i} h^{t_i} = E_0 E_1^i \ldots E_{k-1}^{i^{k-1}}.$$

If, instead, s is computed as $s = as' \bmod q$ for some $a \in \mathbb{Z}_q^*$, then P_i can compute his share (s_i, t_i) and $(E_0, E_1, \ldots, E_{k-1})$ as follows

$$\begin{aligned} E_j &= E_j^a \qquad \text{for } j = 0, 1, \ldots, k-1 \\ s_i &= as_i' \bmod q \\ t_i &= at_i' \bmod q \end{aligned}$$

Again, it is easy to see that

$$g^{s_i} h^{t_i} = E_0 E_1^i \ldots E_{k-1}^{i^{k-1}}.$$

In both of the above cases Lemma 4.2 implies that any k shareholders who have accepted their shares of s' and s'' can find a pair (s, t) such that

$$g^s h^t = E_0.$$

Furthermore, it is an immediate consequence of Theorem 4.4 that fewer than k persons have no information about s if s' and s'' are distributed correctly.

5.2 Choosing an Anonymous Shared Secret

In [IS91] it was shown how to set up a secret sharing scheme without a mutually trusted authority, who knows the secret and distributes it. In this section we show how to achieve the same goal with verifiable secret sharing by demonstrating how n participants can select a secret so that nobody knows it and distribute it verifiably among themselves in a (k, n) secret sharing scheme. It is not hard to generalize the proposed method to let l person $(k \leq l \leq n)$ select and distribute the secret.

Let $P_1, \ldots, P_n$ be the n persons who want to choose a secret and distribute it among themselves and assume that each P_i can make digital signatures. The protocol for P_i is

1. Choose $s_{i0} \in \mathbb{Z}_q$ at random.
2. Distribute s_{i0} verifiably among $P_1, \ldots, P_n$.
   Furthermore P_i signs each secret share and sends the signature with the share.
3. Verify all the received shares. If a share is incorrect, P_i publishes the share and its signature. Then P_i stops.
4. Compute the share (s_i, t_i) of $s = s_{10} + s_{20} + s_{n0}$ and the corresponding public information $(E_0, E_1, \ldots, E_{k-1})$ as described in Subsection 5.1.

It follows from the arguments in the previous subsection that

- (s_i, t_i) is a correct share of s if P_i has accepted all shares correctly; and
- any k participants can find a pair (s', t') such that $E_0 = E(s', t')$.

We now show that s is uniformly distributed in $\mathbb{Z}_q$, and that fewer than k participants have no information about s.

Theorem 5.1
If P_i chooses $s_{i0} \in \mathbb{Z}_q$ uniformly at random and at most $k-1$ of the other parties cooperate, then s is uniformly distributed in $\mathbb{Z}_q$.

Proof
Follows from the fact that no set of at most $k-1$ participants (excluding P_i) get any information about s_{i0}. This implies that if at least one of the participants chooses s_{i0} at random then

$$s = s_{10} + s_{20} + \ldots + s_{n0}$$

is uniformly chosen in $\mathbb{Z}_q$. ∎

As before let $views_S$ be the messages, which the participants in a subset S of $\{1, \ldots, n\}$ see.

Theorem 5.2
For any $S \subset \{1, \ldots, n\}$ of size at most $k-1$ and any $views_S$

$$Prob[s \text{ is chosen} \mid views_S] = \frac{1}{q}$$

for all $s \in \mathbb{Z}_q$, if the participants not in S follow the protocol.

Proof sketch
Under the assumptions in the theorem it follows from Theorem 5.1 that each $s \in \mathbb{Z}_q$ is chosen with probability $\frac{1}{q}$.

Given $S \subset \{1, \ldots, n\}$ of size $k-1$ and $views_S$. For any $s \in \mathbb{Z}_q$ there exists $q^{n-(k-1)-1} = q^{n-k}$ values of $(s_{j0})_{j \notin S}$ such that $s = \sum_{j=1}^{n} s_{j0}$, and as in the proof of Theorem 4.4 for each for these values of s_{j0} $(j \notin S)$ there is exactly one value of t_{j0} which gives the same messages from P_j. ∎

6 Conclusion

We have presented a non-interactive verifiable (k, n)-threshold scheme which is at least as efficient as earlier proposals. Unlike the schemes in [BGW88], [CCD88] and [RB89] this scheme protects the secret to be distributed unconditionally for any value of k $(1 \leq k \leq n)$, but the correctness of the shares depends on the assumption that the dealer cannot find discrete logarithms before the distribution has been completed. This result is optimal because in any non-interactive verifiable secret sharing scheme, which reveals no information about the secret, it is possible for a dishonest dealer to distribute inconsistent shares in nondeterministic polynomial time.

The information rate of the presented scheme is $\frac{1}{2}$ and the distribution of a secret in $\mathbb{Z}_q$ as well as the verification of a share requires at most $2|q|k$ multiplications modulo p.

It was shown that it is very easy to compute linear combinations of shared secrets, and in particular it was demonstrated how, l persons, $P_1, \ldots, P_l$, can select a secret democratically (without knowing the secret) and distribute it verifiably to $P_1, \ldots, P_l$, $P_{l+1}, \ldots, P_n$ in a (k, n)-threshold scheme.

References

[BCC88] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37:156–189, 1988.

[BCP] J. Bos, D. Chaum, and G. Purdy. A voting scheme. Preliminary draft.

[BD90] E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes. In *Advances in Cryptology - proceedings of CRYPTO 89*, pages 278 - 285, 1990.

[Ben87] J. C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In *Advances in Cryptology - proceedings of CRYPTO 86*, Lecture Notes in Computer Science, pages 251–260. Springer-Verlag, 1987.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings AFIPS 1979 Nat. Computer Conf.*, pages 313 - 319, 1979.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal of Computation*, 13:850–864, 1984.

[BGW88] M. Ben-Or, S. Goldwasser, and A. Widgerson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 11–19, 1988.

[Fel87] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, pages 427 - 437, 1987.

[IS91] I. Ingemarsson and G. J. Simmons. A protocol to set up shared secret schemes without the assistance of a mutually trusted party. In *Advances in Cryptology - proceedings of EUROCRYPT 90*, Lecture Notes in Computer Science, pages 266 – 282. Springer-Verlag, 1991.

[Ped91] T. P. Pedersen. Distributed provers with applications to undeniable signatures, 1991. To appear in the proceedings of Eurocrypt'91.

[RB89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing*, pages 73 – 85, 1989.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[Sim90] G. J. Simmons. How to (really) share a secret. In *Advances in Cryptology - proceedings of CRYPTO 88*, Lecture Notes in Computer Science, pages 390 – 448. Springer-Verlag, 1990.

[Wag79] S. S. Wagstaff Jr. Greatest of the least primes in arithmetic progression having a given modulus. *Mathematics of Computation*, 33(147):1073 – 1080, July 1979.