A Verifiable Secret Sharing Scheme with Statistical Zero-Knowledge*

Chunming Tang [†] Zhuojun Liu [‡] Mingsheng Wang[§]

Abstract

In this paper, we first propose a protocol in which the prover can show that $a = b$ holds for two committed integers a and b; also, we present a protocol in which the prover can prove that $a \neq 0$ holds for committed integer a; then, we construct a protocol to prove that the degree of a polynomial $f(x)$ equals to $t-1$ exactly, which has been as an open problem(see[21]); finally, we provide a protocol in which the prover proves that a pair (x, y) is generated by a polynomial $f(x)$, i.e., $y \equiv f(x)(mod\ m)$, where m is a prime.

Based on above four protocols, we put forward a verifiable (t, n)-secret sharing scheme, which can avoid all known the dealer's cheats.

In particular, all above protocols are statistical zero-knowledge.

Keywords: secret sharing, verifiable secret sharing, statistical zero-knowledge

1 Introduction

Informally, a (t, n)-secret sharing is a protocol in which a dealer splits a secret K into n shares, and gives each player a share, where t is called a threshold. This means that $t-1$ or less players can not recover K, but any set of t or more players are guaranteed to easily computes K.

The notion of secret sharing was independently invented by Shamir[1] and Blakey[3], and some other secret sharing schemes were appeared afterwards [4, 5].

*This work was supported in part by 973 project G1998030600.

[†] ctang@mmrc.iss.ac.cn, Key Laboratory of Mathematics Mechanization, Institute of Systems Science, Chinese Academy of Sciences, P.R.China

[‡] zliu@zgc.gov.cn, Key Laboratory of Mathematics Mechanization, Institute of Systems Science, Chinese Academy of Sciences, P.R.China

[§] mswang@is.iscas.ac.cn, State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, P.R.China

According to Shamir scheme, the dealer must select a polynomial $f(x) \in Z_m[x]$ with degree $t-1$, where m is a prime and $K \equiv f(0)(mod\, m)$ is the secret. Each player is given a share (x, y), where x is the public ID of the player and $y \equiv f(x)(mod\, m)$. A secret sharing is *perfect* if no $t-1$ or less than $t-1$ players can not obtain nothing about of the value of K, but at least t players can compute the secret K. The Shamir scheme is perfect, however, Blakey scheme is not.

In some cases, the dealer(or some players) may be bad and deviate from his(or their) prescribed instructions in a (t, n)-secret sharing, such as, the dealer selects a polynomial $f(x)$ with degree not equal to $t-1$, or the dealer gives a player a false share (x, y), i.e., $y \neq f(x)(mod\, m)$, or some players send wrong shares during the secret recovery. In order to prevent these cheat behaviors, some improved methods were proposed[6, 7, 8, 9].

In [9], Benaloh presented a cut-and-choose protocol in which any player can be convinced the dealer's polynomial $f(x)$ with degree at most $t-1$. But, the protocol does not satisfy the following properties: 1) the dealer convinces all players the polynomial $f(x)$ with degree $t-1$ exactly; 2) the protocol is a zero-knowledge proof. In [21], it has already been proposed as an open problem how to construct a protocol satisfying 1) and 2). In this paper, we put forward a protocol satisfying 1) and 2).

Intuitively, a secret sharing is a verifiable secret sharing (VSS for short) if all players believe that their shares are true when they get them from the dealer. Therefore, in a VSS scheme, there is not the case, which the dealer transmit a false share to any player. The notion of VSS was first introduced by Chor, Goldwasser, Micali, and Awerbuch[6], and a constant round interactive scheme for VSS is presented based on the assumed intractability of factorization. The powerful zero-knowledge proof systems of Goldreich, Micali and Wigderson [7] can be used to create a constant round interactive verifiable secret sharing protocol, and their solution may be based on the existence of any one-way function. Their scheme is computational zero-knowledge. Feldman[8] introduced a non-interactive VSS protocol with homomorphic encryption functions, however the protocol has more computation complexity than the interactive VSS protocol.

In this paper, we present a new interactive scheme for VSS. Our scheme satisfies three properties: 1) it is based on the discrete logarithm; 2) any player can check that $y \equiv f(x)(mod\, m)$ even they do not know the polynomial $f(x)$; 3) it is statistical zero-knowledge. As a result, 1) our scheme is more secure than the scheme in [7], because our scheme is statistical zero-knowledge and their scheme computational zero-knowledge; 2) computation complexity of our scheme is less than that of Feldman's scheme which is non-interactive zero-knowledge.

The structure of this paper is following, some preliminaries are introduced in section 2; some basic tools are given in section 3; we generalize the results in [2] in section 4; in section 5 we propose four protocols, i.e., 1) a protocol proving two committed integers being equal, 2) a protocol proving a committed integer $a \neq 0$, 3) a protocol proving a pair (x, y) satisfying $y \equiv f(x)(mod\, m)$ and 4) a protocol proving degree of $f(x)$ being $t-1$ exactly; and a statistical zero-knowledge scheme for VSS is proposed in section 6; finally, concluding remarks will be given in section 7.

2 Preliminaries

2.1 The Network

We consider a network of $n+1$ processors, which consists of n players with identities $1, 2, ..., n$ and a dealer. Each player is a polynomial-time algorithm. We assume that every processor has a broadcast channel; a message sent on such a channel is received by all processors. Additionally, we assume that there is a private channel from each processor to every other processor.

2.2 Zero-knowledge Proof

2.2.1 Indistinguishability

Goldwasser, Micali, and Rackoff[10] defined the notion of perfect(statistical, computational) indistinguishability. Let $L \subset \{0,1\}^*$ be a language.

We call two ensembles $\{U_x\}_{x \in L}$ and $\{V_x\}_{x \in L}$ *perfectly indistinguishable* if $prob(U_{(x)} = \alpha) = prob(V_{(x)} = \alpha)$ holds for each arbitrary size $\alpha \in \{0,1\}^*$.

Two families of random ensembles $\{U_x\}_{x \in L}$ and $\{V_x\}_{x \in L}$ are *statistically indistinguishable* on L if

$$\sum_{\alpha \in \{0,1\}^*} |prob(U_x = \alpha) - prob(V_x = \alpha)| < |x|^{-c}$$

for all constants $c > 0$ and all sufficiently long $x \in L$.

To formalize the notion of computational indistinguishability we make use of non-uniformity. Thus, our judge will be a poly-size family of circuits. That is a family $C = \{C_x\}$ of Boolean circuits C_x with one Boolean output such that, for some constant $e > 0$, all $C_x \in C$ have at most $|x|^e$ gates. In order to feed samples from our probability distributions to such circuits, we will consider only poly-bounded families of random variables, thai is, families $U = \{U(x)\}$ such that, for some constant $d > 0$, all random variable $U(x) \in U$ assigns positive probability only to strings whose lengths are exactly $|x|^d$. If $U = \{U(x)\}$ is a poly-bounded family of random variables that C_x outputs 1 on input a random string distributed according to $U(x)$.

Two poly-bounded families of random variables $\{U_x\}_{x \in L}$ and $\{V_x\}_{x \in L}$ are *computationally indistinguishable* on L if for all poly-size family of circuits C, for all constants $c > 0$ and all sufficiently long strings $x \in L$

$$|prob(U, C, x) - prob(V, C, x)| < |x|^{-c}.$$

2.2.2 Zero-knowledge proof system

Intuitively, a protocol is zero-knowledge if for any adversary A acting on it, there is no a probabilistic polynomial time algorithm(PPTA) which could output strings distinguishable from those then output by A.

We define zero-knowledge for interactive protocol (P, V).

Definition 2.1 *Let P be a probabilistic Turing machine and V a probabilistic polynomial-time Turing machine that share the same input and can communicate with each other. Let L be a language. We say that a pair (P, V) is a perfect(statistical, computational) zero-knowledge proof system for L if*

1. *(Completeness) For all $x \in L$,*

$$Prob[t \leftarrow (P, V)(x); V(x, t) = ACCEPT] = 1.$$

2. *(Soundness) For all $X \notin L$, and any Turing machine P', it holds that*

$$Prob[t \leftarrow (P', V)(x); V(x, t) = ACCEPT] \leq 1/2.$$

3. *(Perfect(statistical, computational zero-knowledge)) For any probability polynomial time algorithm V', there exists a polynomial time algorithm S, called the simulator, such that for all $x \in L$ the following holds:*

   - *$S_{V'}(x) =\perp$(special symbol) with probability at most $1/2$;*
   - *Conditioned on $S_{V'}(x) \neq \perp$, the two distributions $S_{V'}(x)$ and $View_{V'}(x) = \{(r, t) | t \leftarrow (P, V(r))(x)\}$ are perfect(statistical, computational) indistinguishable.*

Remark: An all-powerful adversary can obtain negligible something from prover in statistical zero-knowledge proof, however, it can get some information from computational zero-knowledge proof. Hence, the statistical zero-knowledge has better properties than computational zero-knowledge.

2.3 Verifiable Secret Sharing

2.3.1 Shamir secret sharing

In [1] and [3], Shamir and Blakley presented independently a secret sharing scheme, however, the former is based on the Lagrange interpolation formula and the latter is based on the projective ways of linear geometry. In this paper, we consider mainly the former because the latter is not perfect.

Informally, Shamir (t, n)-secret sharing scheme is the following:

Preliminary: Assume that p is a prime number and the dealer has a secret $K \in Z_p$, all players have a one and only public identity $\in \{1, ..., n\}$.

The secret sharing: The dealer lets $a_0 = K$, chooses independently at random integers $a_1, a_2, ..., a_{t-1} \in Z_p$, and constructs a polynomial $f(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1}$. Assume x is the identity of the i-th player, then the share (x, y), which satisfies $y \equiv f(x)(mod\, p)$, is sent to the i-th player, where all x are public.

The secret reconstructing: When the secret K is reconstructed, at least t players pool their shares, and they can compute easily the secret K according to the Lagrange interpolation formula. Obviously, no less than or equal to $t - 1$ players can recover the secret K.

2.3.2 Verifiable Secret Sharing

Informally, a verifiable secret sharing protocol must meet the following two requirements:

Verifiablility constraint: upon receiving a share of the secret, a player must be able to test whether it is a valid share or not.

Unpredictability: there is no polynomial-time strategy for picking $t-1$ or less than $t - 1$ shares of the secret, such that they can be used to predict the secret with any perceivable advantage.

This framework allows for an interactive protocol proving validity of the shares. Obviously, the same shares are not valid for different secrets; we introduce a zero-knowledge proof which is based on discrete logarithm to deal with it.

3 Basic tools

3.1 Commitment schemes

Pederson[11] proposed a computationally binding and unconditionally hiding scheme based on the discrete logarithm problem. Given a group G of prime order q and two random generators g and h such that $\log_g h$ is unknown and computing discrete logarithms is infeasible. A value $\alpha \in Z_q$ is committed to as $C_\alpha := g^\alpha h^r$, where r is randomly chosen from Z_q. We will use this commitment scheme for our construction and hence they will be statistical zero-knowledge proof of knowledge.

3.2 Zero-knowledge proofs of knowledge about some modular relations

In this section, we mainly review some results from in [2, 15, 16, 22]. Other zero-knowledge proofs of knowledge based on discrete logarithm are referred in [12, 13, 14, 17, 18, 19, 20].

3.2.1 proving that a discrete logarithm lies in a given range

A statistical zero-knowledge protocol proving that a discrete logarithm lies in a given range in [15, 16] was proposed and is denoted by

$$PK\{(\alpha) : y = g^\alpha \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}}\}.$$

In [22], a statistical zero-knowledge protocol for proving $x \in [a, b]$ was proposed, which is denoted $PK\{(\alpha, \beta : c_x = g^\alpha h^\beta \wedge \alpha \in [a, b])\}$.

3.2.2 Proving in statistical zero-knowledge that $a+b \equiv d(mod\ m)$, $ab \equiv d(mod\ m)$ and $a^b \equiv d(mod\ m)$ hold

Let l be an integer such that $-2^l < a, b, d, m < 2^l$ holds and $\varepsilon > 1$ be security parameters. Furthermore, we assume that a group G of order $q > 2^{2\varepsilon l+5}(= 2^{2\ddot{l}+1})$ and two generators g and h are available such that $log_g h$ is not known. This group could for instance be chosen by the prover in which case she would have to prove that she has chosen it correctly. Finally, let the prover's commitments to a, b, d and m be $c_a := g^a h^{r_1}, c_b := g^b h^{r_2}, c_d := g^d h^{r_3}$, and $c_m := g^m h^{r_4}$, where r_1, r_2, r_3, and r_4 are randomly chosen elements of Z_q.

Camenisch and Michels([2]) assume that the verifier has already obtained the commitments c_a, c_b, c_d, and c_m. Then the prover can convince the verifier that $a + b \equiv d(mod\ m)$ holds by running the protocol denoted:

$$\begin{aligned} S_+ := PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \vartheta, \varrho, \lambda) : & \\ c_a = g^\alpha h^\beta \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge c_b = g^\gamma h^\delta \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge & \\ c_d = g^\varepsilon h^\zeta \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge c_m = g^\eta h^\vartheta \wedge -2^{\ddot{l}} < \eta < 2^{\ddot{l}} \wedge & \\ \frac{c_d}{c_a c_b} = c_m^\varrho h^\lambda \wedge -2^{\ddot{l}} < \varrho < 2^{\ddot{l}}\} & \end{aligned}$$

Alternatively, she can convince the verifier that $ab \equiv d(mod\ m)$ holds by running the protocol:

$$\begin{aligned} S_* := PK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\eta,\vartheta,\xi,\rho,\sigma) : \\ c_a = g^\alpha h^\beta \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge c_b = g^\gamma h^\delta \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \\ c_d = g^\varepsilon h^\zeta \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge c_m = g^\eta h^\vartheta \wedge -2^{\ddot{l}} < \eta < 2^{\ddot{l}} \wedge \\ c_d = c_b^\alpha c_m^\rho h^\sigma \wedge -2^{\ddot{l}} < \rho < 2^{\ddot{l}}\}. \end{aligned}$$

At the same time, they presented a protocol in which the prover can convince the verifier that $a^b \equiv d(mod\ m)$ holds for the committed integers without revealing any further information. The protocol is denoted by S_{exp} and is referred in Appendix A. In the following, when denoting a protocol, we will abbreviate the protocol S_{exp} by a clause like to the statement that is proven and assume that the prover send the verifier all necessary commitments; e.g.,

$$PK\{(\alpha,\beta,\gamma,\delta,\varepsilon,\zeta,\theta,\kappa) : c_a = g^\alpha h^\beta \wedge c_b = g^\gamma h^\delta \wedge c_d = g^\varepsilon h^\zeta \wedge$$

$$c_m = g^\theta h^\kappa \wedge (\alpha^\gamma \equiv \varepsilon(mod\ \theta))\}$$

Theorem 3.1 *Let $a, b, d,$ and m be integers that are committed to by the prover as described above, Then all three Potocols S_+, S_*, and S_{exp} are statistical zero-knowledge proofs that $a + b \equiv d(mod\ m)$, $ab \equiv d(mod\ m)$ and $a^b \equiv d(mod\ m)$ hold, respectively.*

Proof: We explain mainly this reason that $a+b \equiv d(mod\ m)$ holds, however, other proofs(including the next theorem) are omitted.

The statistical zero-knowledge claims follows from the statistical zero-knowledgeness of the building blocks.

Let us argue why the modular relations hold. Running the prover with this protocol and using standard techniques, the knowledge extractor can compute integers $\hat{a}, \hat{b}, \hat{d}, \hat{m}, \hat{r_1}, \hat{r_2}, \hat{r_3}, \hat{r_4}$ such that $c_a = g^{\hat{a}} h^{\hat{r_1}}$, $c_b = g^{\hat{b}} h^{\hat{r_2}}$, $c_d = g^{\hat{d}} h^{\hat{r_3}}$, and $c_m = g^{\hat{m}} h^{\hat{r_4}}$ holds. Moreover, $-2^{\ddot{l}} < \hat{a} < 2^{\ddot{l}}$, $-2^{\ddot{l}} < \hat{b} < 2^{\ddot{l}}$, $-2^{\ddot{l}} < \hat{d} < 2^{\ddot{l}}$, and $-2^{\ddot{l}} < \hat{m} < 2^{\ddot{l}}$ holds for these integers.

When running the prover with S_+, the knowledge extractor can further compute integers $\hat{r_5} \in Z_q$ and $\hat{u}$ with $-2^{\ddot{l}} < \hat{u} < 2^{\ddot{l}}$ such that $c_d/(c_a c_b) = c_m^{\hat{u}} h^{\hat{r_5}}$ holds.

Therefore we have $g^{\hat{d}-\hat{a}-\hat{b}} h^{\hat{r_3}-\hat{r_1}-\hat{r_2}} = g^{\hat{m}\hat{u}} h^{\hat{u}\hat{r_4}+\hat{r_5}}$ and hence, provided that the discrete log of h to the base g is not known, we must have

$$\hat{d} \equiv \hat{a} + \hat{b} + \hat{u}\hat{m}(mod\ q).$$

Thus we have $\hat{d} = \hat{a} + \hat{b} + \hat{u}\hat{m} + \bar{w}q$ for some integer $\bar{w}$. Since $2^{2\ddot{l}+1} < q$ and due to the constraints on $\hat{a}, \hat{b}, \hat{d}, \hat{m}, \hat{u}$ we can conclude that the integer $\bar{w}$ must be 0 and hence

$$\hat{d} \equiv \hat{a} + \hat{b} (mod\ \hat{m})$$

must hold. $\blacksquare$

3.2.3 proving the pseudo-primality of a committed number

In [2], J.Camenish and M.Michels show how the prover and the verifier can do Lehmann's primality test[1] for a number committed by prover such that the verifier is convinced that the test was correctly done but does not learn any other information. The general idea is that the prover commits to s random bases a_i and then prove that for these bases $a_i^{(m-1)/2} \equiv \pm 1 (mod\, m)$ holds. Furthermore, the prover must commit to a base, say ã, such that $\tilde{\mathrm{a}}^{(m-1)/2} \equiv -1 (mod\, m)$ holds to satisfy the second condition in Lehmann's primality test. We call this protocol S_{prime} which is described in Appendix B. In the following section, $PK : \{(\alpha, \beta) : c_m = g^\alpha h^\beta \wedge \alpha \in \{prime\}\}$ denotes proving that an integer m is a prime by S_{prime}.

Theorem 3.2 *Given a commitment c_m to an integer, the protocol S_{prime} is a statistical zero-knowledge proof that the committed integer is a prime with error-probability at most 2^{-s} for the primality-test.*

All described protocols can be combined in natural ways. First of all, one can use multiple bases instead of a single one in any of the above proofs. Then, executing any number of instances of these protocols in parallel and choosing the same challenges for all of them in each round corresponding to the $\wedge$-composition of the statements the single protocols prove.

4 The statistical zero-knowledge proof for $a+b = d$, $ab = d$, and $d = a^b$

In this section, we will generalize these results in 3.2.2 and construct the statistical zero-knowledge proof for $a + b = d$, $ab = d$, and $d = a^b$, furthermore, the verifier also obtains nothing information except commitments to some integers.

[1] An odd integer $m > 1$ is *prime* if and only if

$$\forall a \in Z_m^* : \quad a^{(m-1)/2} \equiv \pm 1 \ (mod\ m) \ and \ \exists a \in Z_m^* : \quad a^{(m-1)/2} \equiv -1 \ (mod\ m).$$

Assume l, q and commitment scheme be uniform in 3.2.2, and the verifier gets commitments c_a, c_b, c_d to a, b, d, respectively. Then, in the following two protocols S'_+ and S'_* the prover can convince the verifier that $a + b = d$ and $ab = d$ hold.

$$\begin{aligned}
S'_+ &:= PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \lambda) : \\
&\quad c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
&\quad c_b = g^{\gamma} h^{\delta} \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \\
&\quad c_d = g^{\varepsilon} h^{\zeta} \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge \\
&\quad \tfrac{c_d}{c_a c_b} = h^{\lambda}\}
\end{aligned}$$

$$\begin{aligned}
S'_* &:= PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \sigma) : \\
&\quad c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
&\quad c_b = g^{\gamma} h^{\delta} \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \\
&\quad c_d = g^{\varepsilon} h^{\zeta} \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge \\
&\quad c_d = c_b^{\alpha} h^{\sigma}\}
\end{aligned}$$

The following protocol S'_{exp} will guarantee that the prover convinces the verifier that $a^b = d$ holds.

$$S'_{exp} := PK\{(\alpha, \beta, \xi, \chi, \gamma, \delta, \eta, (\lambda_i, \mu_i, \xi_i, \sigma_i, \tau_i, \vartheta_i, \psi_i)_{i=1}^{l_b-1}, (\omega_i, \rho_i)_{i=1}^{l_b-2},) :$$

$$\begin{gathered}
c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
c_d = g^{\gamma} h^{\delta} \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \\
(\textstyle\prod_{i=0}^{l_b-1} c_{b_i}^{2^i})/c_b = h^{\eta} \wedge \\
c_{v_1} = g^{\lambda_1} h^{\mu_1} \wedge ... \wedge c_{v_{l_b-1}} = g^{\lambda_{l_b-1}} h^{\mu_{l_b-1}} \wedge \\
c_{v_1} = c_a^{\alpha} h^{\xi_1} \wedge c_{v_2} = c_{v_1}^{\lambda_1} h^{\xi_2} \wedge ... \wedge c_{v_{l_b-1}} = c_{\nu_{l_b-2}}^{\lambda_{l_b-2}} h^{\xi_{l_b-1}} \wedge \\
-2^{\ddot{l}} < \lambda_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \lambda_{l_b-1} < 2^{\ddot{l}} \wedge \\
c_{\mu_1} = g^{\omega_1} h^{\rho_1} \wedge ... \wedge c_{\mu_{l_b-2}} = g^{\omega_{l_b-2}} h^{\rho_{l_b-2}} \wedge \\
-2^{\ddot{l}} < \omega_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \omega_{l_b-2} < 2^{\ddot{l}} \wedge \\
((c_{b_0} = h^{\sigma_0} \wedge c_{\mu_0}/g = h^{\tau_0}) \vee (c_{b_0}/g = h^{\vartheta_0} \wedge c_{\mu_0}/c_a = h^{\psi_0})) \wedge \\
((c_{b_1} = h^{\sigma_1} \wedge c_{\mu_1}/c_{\mu_0} = h^{\tau_1}) \vee \\
(c_{b_1}/g = h^{\vartheta_1} \wedge c_{\mu_1} = c_{\mu_0}^{\lambda_1} h^{\psi_1})) \wedge ... \wedge \\
((c_{b_{l_b-2}} = h^{\sigma_{l_b-2}} \wedge c_{\mu_{l_b-2}}/c_{\mu_{l_b-3}} = h^{\tau_{l_b-2}}) \vee \\
(c_{b_{l_b-2}}/g = h^{\vartheta_{l_b-2}} \wedge c_{\mu_{l_b-2}} = c_{\mu_{l_b-3}}^{\lambda_{l_b-2}} h^{\psi_{l_b-2}})) \wedge \\
((c_{b_{l_b-1}} = h^{\sigma_{l_b-1}} \wedge c_d/c_{\mu_{l_b-2}} = h^{\tau_{l_b-1}}) \vee \\
(c_{b_{l_b-1}}/g = h^{\vartheta_{l_b-1}} \wedge c_d = c_{\mu_{l_b-2}}^{\lambda_{l_b-1}} h^{\psi_{l_b-1}}))\}
\end{gathered}$$

Theorem 4.1 *Let a, b, and d be integers that are committed to by the prover as described above, Then all three Protocols S'_+, S'_*, and S'_{exp} are statistical zero-knowledge proofs that $a + b = d$, $ab = d =$ and $a^b = d$ hold, respectively.*

Proof: We explain mainly this reason that $a + b = d$ holds, however, the proofs of $ab = d$ and $a^b = d$ are omitted.

The statistical zero-knowledge claims follows from the statistical zero-knowledgeness of the building blocks.

Running the prover with this protocol and using standard techniques, the knowledge extractor can compute integers $\hat{a}, \hat{b}, \hat{d}, \hat{r_1}, \hat{r_2}, \hat{r_3}$ such that $c_a = g^{\hat{a}} h^{\hat{r_1}}$, $c_b = g^{\hat{b}} h^{\hat{r_2}}$, and $c_d = g^{\hat{d}} h^{\hat{r_3}}$ hold. Moreover, $-2^{\ddot{l}} < \hat{a} < 2^{\ddot{l}}$, $-2^{\ddot{l}} < \hat{b} < 2^{\ddot{l}}$, and $-2^{\ddot{l}} < \hat{d} < 2^{\ddot{l}}$, hold for these integers.

When running the prover with S'_+, the knowledge extractor can further compute integers $\hat{r_4} \in Z_q$ such that $c_d/(c_a c_b) = h^{\hat{r_4}}$ holds.

Therefore we have $g^{\hat{d}-\hat{a}-\hat{b}} h^{\hat{r_3}-\hat{r_1}-\hat{r_2}} = h^{\hat{r_5}}$ and hence, provided that the discrete log of h to the base g is not known, we must have

$$\hat{d} \equiv \hat{a} + \hat{b} (mod\ q).$$

Thus we have $\hat{d} = \hat{a} + \hat{b} + \bar{w} q$ for some integer $\bar{w}$. Since $2^{2\ddot{l}+1} < q$ and due to the constraints on $\hat{a}, \hat{b}, \hat{d}$ we can conclude that the integer $\bar{w}$ must be 0 and hence

$$\hat{d} = \hat{a} + \hat{b}$$

must hold. $\blacksquare$

In the following, when denoting a protocol, we will abbreviate the protocol S'_{exp} by a clause like to the statement that is proven and assume that the prover send the verifier all necessary commitments; e.g.,

$$PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta) : c_a = g^{\alpha} h^{\beta} \wedge c_b = g^{\gamma} h^{\delta} \wedge c_d = g^{\varepsilon} h^{\zeta} \wedge (\alpha^{\gamma} = \varepsilon)\}$$

Remark: By using protocol S'_+, S'_*, we can construct a statistical zero-knowledge proof proving that a committed integer a is either odd or even.

5 Several practical statistical zero-knowledge proof protocols

In this section, we construct several statistical zero-knowledge proofs in order to construct a statistical zero-knowledge scheme for VSS in the next section.

5.1 Proving that two committed integers are equal

Assume c_a and c_b are commitments to integers a and b, and the verifier has obtained these commitments before the protocol beginning. A protocol, in which the prover convinces the verifier that $a = b$ holds, will be proposed in this section, and it is denoted by $S_=$.

$$S_= : PK\{(\alpha, \beta, \gamma, \delta, \lambda) :$$

$$c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \tag{1}$$

$$c_b = g^{\gamma} h^{\delta} \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \tag{2}$$

$$\frac{c_a}{c_b} = h^{\lambda}\} \tag{3}$$

Theorem 5.1 *If c_a and c_b are commitments to integers a and b as described above, the protocol $S_=$ is a statistical zero-knowledge proof that $a = b$ holds.*

Proof: The statistical zero-knowledge claims follows from the statistical zero-knowledgeness of commitment scheme.

Running the prover with this protocol and using standard techniques, the knowledge extractor can compute integers $\hat{a}, \hat{b}, \hat{r_1}, \hat{r_2}$ such that $c_a = g^{\hat{a}} h^{\hat{r_1}}$ and $c_b = g^{\hat{b}} h^{\hat{r_2}}$, hold. Moreover, $-2^{\ddot{l}} < \hat{a} < 2^{\ddot{l}}$, and $-2^{\ddot{l}} < \hat{b} < 2^{\ddot{l}}$ hold for these integers.

When running the prover with $S_=$, the knowledge extractor can further compute integers $\hat{r_3} \in Z_q$ such that $c_a/c_b = h^{\hat{r_3}}$ holds.

Therefore we have $g^{\hat{a}-\hat{b}} h^{\hat{r_1}-\hat{r_2}} = h^{\hat{r_3}}$ and hence, provided that the discrete log of h to the base g is not known, we must have

$$\hat{a} \equiv \hat{b} (mod\ q).$$

Thus we have $\hat{a} = \hat{b} + \bar{w}q$ for some integer $\bar{w}$. Since $2^{2\ddot{l}+1} < q$ and due to the constraints on $\hat{a}, \hat{b}$ we can conclude that the integer $\bar{w}$ must be 0 and hence

$$\hat{a} = \hat{b}$$

must hold. ∎

Assume an integer a is known, the following protocol $S'_=$ is a statistical zero-knowledge proof that the committed integer b is equal to a.

$$S'_= : PK\{(\alpha, \beta, \lambda) :$$

$$c_b = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \tag{4}$$

$$\frac{c_b}{g^a} = h^{\lambda}\} \tag{5}$$

5.2 Proving that a committed integer is not equal to 0

In this section, we will present a protocol by which the prover can convince the verifier that an integer a is not 0, furthermore, it is statistical zero-knowledge.

For an arbitrary integer a, it can be written $\prod_{i=1}^{i=t} p_i^{k_i}$, where $p_1, ..., p_r$ are primes and $k_1, ..., k_r$ are integers. Now, if the prover can prove that a has form $\prod_{i=1}^{r} p_i^{k_i}$ and all $p_1, ..., p_r$ are primes, then $a \neq 0$ holds.

Assume l, q and commitment scheme be set in 3.2.2, and let prover's commitments to $a, s_1 = p_1^{k_1}, ..., s_r = p_r^{k_r}, p_1, ..., p_r, k_1, ..., k_r$, and suppose the verifier has already obtained all commitments before the protocol begins. The following protocol will prove that the integer a is not 0.

$S_{a \neq 0} := PK\{(\alpha, \beta, \rho, (\delta_i, \varepsilon_i, \zeta_i, \eta_i, \theta_i, \mu_i)_{i=1}^{i=r}):$

$$c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \tag{6}$$

$$c_{s_1} = g^{\delta_1} h^{\varepsilon_1} \wedge ... \wedge c_{s_r} = g^{\delta_r} h^{\varepsilon_r} \wedge \tag{7}$$

$$(-2^{\ddot{l}} < \delta_1 < 2^{\ddot{l}}) \wedge ... \wedge (-2^{\ddot{l}} < \delta_r < 2^{\ddot{l}}) \wedge \tag{8}$$

$$c_a / c_{s_1} ... c_{s_r} = h^{\rho} \wedge \tag{9}$$

$$c_{p_1} = g^{\zeta_1} h^{\eta_1} \wedge ... \wedge c_{p_r} = g^{\zeta_r} h^{\eta_r} \wedge \tag{10}$$

$$(-2^{\ddot{l}} < \zeta_1 < 2^{\ddot{l}}) \wedge ... \wedge (-2^{\ddot{l}} < \zeta_r < 2^{\ddot{l}}) \wedge \tag{11}$$

$$c_{k_1} = g^{\theta_1} h^{\mu_1} \wedge ... \wedge c_{k_r} = g^{\theta_r} h^{\mu_r} \wedge \tag{12}$$

$$(-2^{\ddot{l}} < \theta_1 < 2^{\ddot{l}}) \wedge ... \wedge (-2^{\ddot{l}} < \theta_r < 2^{\ddot{l}}) \wedge \tag{13}$$

$$(\delta_1 = \zeta_1^{\theta_1}) \wedge ... \wedge (\delta_r = \zeta_r^{\theta_r}) \wedge \tag{14}$$

$$(\zeta_1 \in \{prime\}) \wedge ... \wedge (\zeta_r \in \{prime\})\} \tag{15}$$

Theorem 5.2 *Let a be an integer that is committed by c_a. Then $S_{a \neq 0}$ is a statistical zero-knowledge proof that $a \neq 0$ holds.*

Proof: *Completeness:* If $a \neq 0$, the prover can prove that $a = \prod_{i=1}^{r} p_i^{k_i}$ holds in (6)-(14); in (15), the prover proves that all of $p_1, ..., p_r$ are prime numbers. As a result, the verifier believes that $a \neq 0$ holds .

Soundness: If $a = 0$, the prover may prove that a is a composite integer in (6)-(14); however, she can not prove that each of $p_1, ..., p_r$ is prime; so, the verifier rejects.

Zero-knowledgeness: $S_{a \neq 0}$ is statistical zero-knowledge from Theorem 4.1. ▮

5.3 Proving a polynomial $f(x)$ with degree $t-1$ exactly

Assume $c_b = (c_{b_0}, c_{b_1}, ..., c_{b_m})$ and $c_a = (c_{a_0}, c_{a_1}, ..., c_{a_m})$ are commitments to all exponents of x and all coefficients, respectively, in polynomial $f(x)$, furthermore, we assume that the i-th term is $a_i x^{b_i}$, that is, $f(x) = a_0 x^{b_0} +$

$a_1 x^{b_1} + ... + a_m x^{b_m}$. If all above commitments satisfy the following: 1) there exists a committed integer b_j is equal to $t-1$; 2) other all committed integers $b_i \in [0, t-2]$, where $i \neq j$; 3) $a_j \neq 0$ holds, then the degree of the polynomial $f(x)$ is $t-1$ exactly. In total subsection, we assume arbitrary two committed integers b_i and b_k is not equal, where $i \neq k$ and $m \leq r$.

Protcol 1

1. the prover chooses randomly a permutation π, obtains two new vectors $c_{a'} = \pi c_a = (c_{a'_0}, c_{a'_1}, ..., c_{a'_m})$ and $c_{b'} = \pi c_b = (c_{b'_0}, c_{b'_1}, ..., c_{b'_m})$, and sends $c_{a'}$ and $c_{b'}$ to the verifier.

2. The prover proves to the verifier that a committed integer $b'_j = t - 1$ holds by $S'_=$.

3. The prover proves to the verifier that the committed integer $a'_j \neq 0$ holds by $S_{a \neq 0}$.

4. The prover proves to the verifier that all committed integer $b'_i \in [0, t-2]$, where $i \neq j$, i.e., $PK : \{((\alpha_i, \beta_i)_{i=0, i \neq j}^m) : c_{b'_0} = g^{\alpha_0} h^{\beta_0} \wedge \alpha_0 \in [0, t-2] \wedge ... \wedge c_{b'_{j-1}} = g^{\alpha_{j-1}} h^{\beta_{j-1}} \wedge \alpha_{j-1} \in [0, t-2] \wedge c_{b'_{j+1}} = g^{\alpha_{j+1}} h^{\beta_{j+1}} \wedge \alpha_{j+1} \in [0, t-2] \wedge ... \wedge c_{b'_m} = g^{\alpha_m} h^{\beta_m} \wedge \alpha_m \in [0, t-2]\}$.

5. The prover obtains the primitive c_a and c_b by $c_a = c'_a \pi^-$ and $c_b = c'_b \pi^-$.

Theorem 5.3 *Let $c_b = \{c_{b_0}, c_{b_1}, ..., c_{b_m}\}$ and $c_a = \{c_{a_0}, c_{a_1}, ..., c_{a_m}\}$ are commitments to all exponents of x and all coefficients, respectively, in polynomial $f(x)$, then the Protocol 1 is a statistical zero-knowledge proof that the degree of the polynomial $f(x)$ is $t-1$ exactly.*

Proof: 1) *Completeness:* If $f(x) = \sum_{i=0}^m a_i x^{b_i}$ is a polynomial with degree $t-1$ exactly, then, in the above protocol the prover can convince the verifier the polynomial with degree $t-1$ exactly. In particular, the verifier does not know j, which satisfies $b_j = t-1$, because we rearrange the c_a and c_b by a random permutation π in the first step in *protocol* 1.

2)*Soundness:* If $f(x)$ is not a polynomial with degree $t-1$ exactly, there exist three cases: 1) if there is not a committed $b_j = t-1$, the second step is wrong; 2) if there is a $b_j = t-1$, however, $a_j = 0$ holds, the third step is wrong; 3) if $b_j = t-1$ and $a_j \neq 0$ hold, but there exists a committed b_k not in $[0, t-2]$, then fourth step is wrong. So, if $f(x)$ is not a polynomial with degree $t-1$ exactly, the prover can convince the verifier the polynomial with degree $t-1$ exactly with negligible probability.

3) *Zero-knowledge:* By $S_{a\neq 0}$, $S'_{=}$ and the result in [22] we know that our protocol satisfies statistical zero-knowledge.

Remark: In an ordinary (t, n)-secret sharing scheme, if the dealer produces the share with a polynomial $f(x)$ with the degree less than $t-1$, however, the verifier doesn't know it and thinks his share generated by a polynomial $f(x)$ with degree $t-1$, then less than $t-1$ players will obtain the secret, as a result, the secret sharing will not be perfect. Now, we can utilize the Protocol 1 to prevent this case.

5.4 The protocol which prove that a point (x, y) satisfies $y \equiv f(x)(mod\, m)$

In an ordinary (t, n)-secret sharing scheme, if the dealer tries to cheat all players with a polynomial $f(x)$, whose degree does not equal to $t-1$, we can overcome it by Protocol 1. But, if the the dealer sends the share (x, y), which does not satisfy $y \equiv f(x)(mod\, m)$, she can still cheat all players, because each player does not know the polynomial $f(x)$ and can't check whether his share (x, y) satisfies $y \equiv f(x)(mod\, m)$ or not.

In this subsection, we present a protocol by which the prover can convince the verifier that (x, y) is produced by the equation

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \equiv y \;\; mod\; m$$

where all integers $x, a_0, ..., a_{t-1}, y, m$ might only given as commitments can be shown.

Assume the algebraic setting be similar to 3.2.2, then the following protocol will guarantee that (x, y) is correctly generated by polynomial $f(x)$, i.e., $y \equiv f(x)(mod\, m)$, however the verifier know nothing but commitments to the polynomial.

Protocol 2

P1: *The prover commits to all the summands* a_0, $s_1 :\equiv a_1 x(mod\, m)$,...,$s_{t-1} :\equiv a_{t-1}x^{t-1}(mod\, m)$ *and shows that* $a_0 + s_1 + ... + s_{t-1} \equiv y(mod\, m)$, *i.e., assume that the verifier already obtained the commitments* $c_{a_0}, c_{s_1}, ..., c_{s_{t-1}}, c_y$, *and* c_m, *then the prover can convince the verifier that* $a_0 + s_1 + ... + s_{t-1} \equiv y(mod\, m)$ *holds.*

V1: *If P1 is real, go on, else, reject.*

P2: *The prover commits to all terms a_i, $p_i \equiv x^i (mod\, m)$ and shows $s_i \equiv a_i p_i (mod\, m)$(let $S_{(*,i)}$ denote the proof of $s_i \equiv a_i p_i (mod\, m)$), where $0 \leq i \leq t-1$, i.e., the prover runs S_* for i times.*
V2: *If all $S_{(*,i)}$ in P2 are real, go on, else, reject.*
P3: *The prover commits to x, all terms i and shows $p_i \equiv x^i (mod\, m)$(let $S_{(exp,i)}$ denote the proof of $p_i \equiv x^i (mod\, m)$), where $0 \leq i \leq t-1$, i.e., the prover runs S_{exp} for $t-1$ times.*
V3: *If all $S_{(exp,i)}$ are real, end, else, reject.*

$$\begin{aligned}
S_{y \equiv f(x)(mod\ m)} := PK\{&(\alpha, \beta, (\alpha_i, \beta_i, \gamma_i, \delta_i, \mu_i, \nu_i, \tau_i, \phi_i)_{i=1}^{t-1}, \varepsilon, \zeta, \eta, \vartheta, \mu, \lambda) : \\
&c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
&c_{s_1} = g^{\gamma_1} h^{\delta_1} \wedge ... \wedge c_{s_{t-1}} = g^{\gamma_{t-1}} h^{\delta_{t-1}} \wedge \\
&-2^{\ddot{l}} < \gamma_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \gamma_{t-1} < 2^{\ddot{l}} \wedge \\
&c_y = g^{\varepsilon} h^{\zeta} \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge \\
&c_m = g^{\eta} h^{\vartheta} \wedge -2^{\ddot{l}} < \eta < 2^{\ddot{l}} \wedge \\
&\frac{c_y}{c_a c_{s_1} ... c_{s_{t-1}}} = c_m^{\mu} h^{\lambda} \wedge -2^{\ddot{l}} < \mu < 2^{\ddot{l}} \wedge \\
&c_{a_1} = g^{\alpha_1} h^{\beta_1} \wedge ... \wedge c_{a_{t-1}} = g^{\alpha_{t-1}} h^{\beta_{t-1}} \wedge \\
&-2^{\ddot{l}} < \alpha_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \alpha_{t-1} < 2^{\ddot{l}} \wedge \\
&c_{p_1} = g^{\mu_1} h^{\nu_1} \wedge ... \wedge c_{p_{t-1}} = g^{\mu_{t-1}} h^{\nu_{t-1}} \wedge \\
&-2^{\ddot{l}} < \mu_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \mu_{t-1} < 2^{\ddot{l}} \wedge \\
&c_{s_1} = c_{p_1}^{\alpha_1} c_m^{\tau_1} h^{\phi_i} \wedge ... \wedge c_{s_{t-1}} = c_{p_{t-1}}^{\alpha_{t-1}} c_m^{\tau_{t-1}} h^{\phi_{t-1}} \wedge \\
&-2^{\ddot{l}} < \tau_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \tau_{t-1} < 2^{\ddot{l}} \wedge \\
&c_x = g^{\varphi} h^{\psi} \wedge -2^{\ddot{l}} < \varphi < 2^{\ddot{l}} \wedge \\
&c_1 = g^{\varphi_1} h^{\psi_1} \wedge ... \wedge c_{t-1} = g^{\varphi_{t-1}} h^{\psi_{t-1}} \wedge \\
&-2^{\ddot{l}} < \varphi_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \varphi_{t-1} < 2^{\ddot{l}} \wedge \\
&(\mu_1 \equiv \varphi^1 (mod\ m) \wedge ... \wedge (\mu_{t-1} \equiv \varphi^{t-1} (mod\ m)\}
\end{aligned}$$

Theorem 5.4 *Let $x, y, a_0, ..., a_{t-1}, 1, ..., t-1$, and m be integers that are committed to by the prover as described in subsection 3.2.2. Then the Protocol 2 is a statistical zero-knowledge proof that $y \equiv a_0 + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1} (mod\, m)$ holds.*

Proof: In P1, the prover can convince the verifier that $y \equiv a_0 + s_1 + s_2 + ... + s_{t-1} (mod\, m)$ holds; In P2, the prover can convince the verifier that $s_i \equiv a_i p_i (mod\, m)$ holds, where $1 \leq i \leq t-1$; In P3, the prover can convince the verifier that $p_i \equiv x^i (mod\, m)$, where $1 \leq i \leq t-1$. So the prover can convince the verifier that $y \equiv a_0 + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1} (mod\, m)$. Obviously, P1, P2, and P3 are statistical zero-knowledge proofs because of the statistical zero-knowledge proof properties of S_+, S_*, and S_{exp}. ▌

6 A verifiable secret sharing scheme

In this subsection, we will construct a verifiable (t, n)-secret sharing scheme.

6.1 The VSS scheme

Assume that the dealer has a secret K. Now, she wants to split it to n players so that any set of less than t players can't recover the secret K, however, any set of u players are guaranteed that they can easily computes K, where $t \leq u \leq n$.

6.1.1 Initialization

Let $Generator(k)$ be a prime number generator, where k is bit number of prime number. Assume $p \leftarrow Generator(l)$, and $q \leftarrow Generator(\ddot{l})$, where $\ddot{l} > 2\varepsilon l + 1$ and ε is a security parameter. $K \in Z_p$ is the secret of the dealer.

6.1.2 The Secret Sharing scheme

D1: *The dealer randomly selects* $t-1$ *integers* $a_1, ..., a_{t-1}$ *in* Z_p*, and lets* $a_0 = K$*, constructs polynomial* $f(x) = a_0 + a_1x + a_2x^2 + ... + a_{t-1}x^{t-1}$*. Then, she commits to* $0, 1, 2, ..., t-1, a_0, a_1, ..., a_{t-1}, p$*, and broadcasts their commitments to all players in broadcast channel.*
D↔P(1): *The dealer runs Protocol 1 with all players.*
P1: *If Protocol 1 is real to each player(i.e., the polynomial* $f(x)$ *with degree exact* $t-1$*), go on, else, reject.*
D2: *For each player, the dealer generates the share* (x_i, y_i)*(i.e.,* $f(x_i) \equiv y_i (mod\, p)$*), where* x_i *is the identity of the i-th player.*
D↔P(2): *The dealer runs* $S_{f(x_i) \equiv y_i}$ *with the i-th player, and sends commitments(to* $x_i, x_i^2, ..., x_i^{t-1}, y_i$*) the i-th player.*
P2: *If* $S_{f(x_i) \equiv y_i}$ *is real, go on, else, reject.*
D3: *The dealer decommits to commitments (to* (x_i, y_i)*).*
P3: *If the dealer can decommit it correctly, the i-th player accepts the share* (x_i, y_i)*, else, reject.*
Remark: The prime q is used in D↔P(1) and D↔P(2).

6.2 Properties of our protocol

Our protocol has the following properties.

6.2.1 It is a verifiable (t, n)-secret sharing

In **D1** and **D2**, the dealer generates the share (x_i, y_i) with the polynomial $f(x) = K + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}$, where $a_1, a_2, ..., a_{t-1}$ and $x \in Z_p$, and K is the secret. It is obvious that any set of less than t players can't recover the secret K because any solutions less than t can't obtain all coefficients $K, a_1, ..., a_{t-1}$ in this polynomial. However, any solutions of $u(\geq t)$ can get the secret K according to Lagrange interpolation formula. So, it is a (t, n)-secret sharing scheme. Obviously, it is verifiable because the player can decide his share (x, y) which satisfies $y \equiv f(x)(mod\ m)$ or not.

6.2.2 it is a statistical zero-knowledge proof

D↔P(1) is Protocol 1, i.e., the dealer explains to all players that degree of her polynomial equals to $t - 1$, so it is statistical zero-knowledge.

D↔P(2) is Protocol 2, i.e., the dealer proves to each player whose share (x_i, y_i) is exactly generated by the polynomial $f(x) = K + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}$, Obviously, it is statistical zero-knowledge.

In Protocol 4, because D↔P(1) and D↔P(2) are statistical zero-knowledge, furthermore, in other steps, each player knows nothing but that the share (x_i, y_i) is generated by the polynomial $f(x) = K + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}$. Hence it is still statistical zero-knowledge.

6.3 Security of our protocol

Our protocol can prevent all known the dealer's cheats. As usually, there exist three type dealer's cheats, one is that the dealer tells all players that her threshold is t, however, the degree of her selected polynomial $f(x)$ does not equal to $t - 1$; the second is that she sends the share (x, y) to a player, but $y \equiv f(x)$ does not hold, then the secret recovered will not be K when at least t players recover the secret; the last is that the dealer claims a secret K, however a_0 in $f(x)$ is different from this secret. The following section will analysis how our protocol can avoid these cases.

6.3.1 a polynomial $f(x)$ whose degree is not $t - 1$

Assume the dealer uses a polynomial $f(x)$, whose degree does not equal to $t - 1$, the D↔P(1) in secret sharing scheme will be unsuccessful, because this step is that the dealer proves to each player her polynomial with degree $t - 1$ exactly.

6.3.2 the share (x, y), sent by the dealer, does not satisfy $y \equiv f(x)(mod\, m)$

Assume the dealer sends the (x, y) to the player, however, y is not generated by the polynomial $f(x)$ in x, i.e., $f(x) \neq y$. As a result, there exists a contradiction because in D↔P(2) in secret sharing scheme is that the dealer proves to the player that $y \equiv f(x)(mod\, m)$.

6.3.3 The secret K claimed by the dealer is different from the a_0(real secret) in polynomial $f(x)$

Assume the dealer uses a secret K_1 in polynomial $f(x)$, however, she claims that her secret is K. When at least t players recover the secret, they get the secret K_1, which is different from K, then all players think that the dealer is cheating. But, the dealer may speak that all players are cheating her(i.e., all players gives a secret different from one recovered by them). For this case, we may ask the dealer to decommit the commitment to K_1 in D1, if the dealer can do it correctly, the dealer is cheating.

Remarks: Our protocol can avoid all known dealer's cheats, however, there exist some players' cheats while reconstructing the secret K, for example, some players provide some false shares. In order to prevent this case, we advance an improved scheme, which is followed:

In initialization in 6.1.1, the dealer chooses a private key e, and publish a public key d, such as RSA system;

In Protocol 4, we add two steps at the last:

D4: the dealer sends encryption E to share (x_i, y_i) by using RSA with secret key e to P_i.

S4: the player P_i decryption to E with public key d, if the value is equal to (x_i, y_i), accepts, else, rejects.

The advanced scheme can avoid this cheat which any player presents a faulty share when he sends his share to compute the secret K, if he is requested to send $(E, (x_i, y_i))$. Because any player can decrypt E with the public key d, if the decryption value is not equal to (x_i, y_i), the i-th player is cheating, else, accepts.

7 Results

In this paper, we first generalize the results in [2], obtain three statistical zero-knowledge proofs in which the prover can convince the verifier that

$a + b = d$, $ab = d$ and $a^b = d$ hold for committed integers a, b, d; then, we propose four statistical zero-knowledge proof; finally, based on all above protocols, we present a verifiable (t, n)-secret sharing scheme, which can prevent all known the dealer's cheats, furthermore, it is statistical zero-knowledge too.

References

[1] A Shamir. How to Share a Secret. *Comm. ACM* 1979, 22: 612-613.

[2] J Camenisch, and M Michels, Proving in Zero-knowledge that a Number is the Product of Two Safe Primes. *BRICS Report Series*, RS-98-29.

[3] G.R Blakey, Safeguarding Cryptographic Keys. *Proc. NCC*, AFIPS Press, Montvale, 1979, 48: 313-317.

[4] G.I Davida, R.A Demillo, and R.J Lipton, Protecting Shared Cryptographic Keys. *Proc. Symp. on Security and Privacy*, IEEE Computer Soc. Press, Silver Spring, MD, April pp.14-16 1980.

[5] S.C Kothari. Generalized Linear Threshold Scheme, *Proc of CRYPTO'84* pp 231-242, Berlin: Springer Verlag, 1984.

[6] B Chor, S Goldwasser, S Micali, and A Awerbuch, Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults, *Proceedings of the 26 IEEE Symposium on Foundation of Computer Science (FOCS), IEEE*, pp. 383–395, 1985.

[7] O Goldreich, S Micali and A Wigderson, Proofs that Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design, *Proceedings of the 27 IEEE Symposium on Foundation of Computer Science (FOCS), IEEE*, pp. 174–187, 1986.

[8] P Feldman, A Practical Scheme for Non-interactive Verifiable Secret Sharing. *Proceedings of the 28 IEEE Symposium on Foundation of Computer Science (FOCS), IEEE*, pp. 427–437, 1987.

[9] J.C Benaloh, Secret Sharing Homomorphisms: Keeping Shares of a Secret. *Proc of CRYPTO'86*, Berlin: Springer, 1986.

[10] S Goldwasser, S Micali, and C Rackoff, The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18,pp 186-208, 1989. Preliminary version in 17th STOC, 1985.

[11] T.P Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing. *Advances in Cryptology-CRYPTO'91*, pp 129-140, Berlin: Springer, 1991.

[12] D Chaum, J.H Evertse, and J van de Graaf, and Peralta R, Demonstrating possession of a discrete logarithm without revealing it. *Advances in Cryptology-CRYPTO'86*, pp 200-212, Berlin: Springer, 1987.

[13] C.P Schnorr, Efficient signature generation for smart cards. *J of Cryptology*, 4(3):239-252, 1991.

[14] J Camenisch, and M Stadler, Efficient group signature schemes for large groups. *Advances in Cryptology-CRYPTO'97*, pp 410-424, Berlin: Springer, 1997.

[15] A Chan, Y Frankel, and Y Tsiounis, Easy come-easy go divisible cash. *Advances in Cryptology-EUROCRYPT'98*, pp 561-575, Berlin: Springer, 1998.

[16] E Fujisaki, and T Okamoto, Statistical zero-knowledge protocols to prove modular polynomial relations. *Advances in Cryptology-CRYPTO'97*, pp 16-30, Berlin: Springer, 1997.

[17] S Brands, Electronic cash systems based on the representation problem in groups of prime order, *Advances in Cryptology-CRYPTO'93*, pp 1-15, Berlin: Springer, 1993.

[18] D Chaum, J.E Evertse, and J van de Graaf, An improved protocol for demonstrating possession of discrete logarithms and some generalizations, *Advances in Cryptology-EUROCRYPT'87*, pp 127-141, Berlin: Springer, 1988.

[19] D Chaum, and T.P Pedersen, Wallet databases with observers, *Advances in Cryptology-CRYPTO'92*, pp 89-105, Berlin: Springer, 1993.

[20] R Cramer, I Damgard, and B Schoenmakers, Proofs of partial knowledge and simplified design of witness hiding protocols, *Advances in Cryptology-CRYPTO'94*, pp 174-187, Berlin: Springer, 1994.

[21] L Xishong, H Liang, and Z Zhencheng, Computer Cryptography and Its Applications, *Industry Publish of National Defence*, Beijing, 2001.

[22] F.Boudot, Efficient Proofs that a Committed Number Lies in an Interval. *Advances in Cryptology-EUROCRYPT'00*, pp 431-444, Berlin: Springer, 2000.

Appendix A

This Protocol will prove that $a^b \equiv d(mod\, m)$ holds.

$S_{exp} := PK\{(\alpha, \beta, \xi, \chi, \gamma, \delta, \varepsilon, \zeta, \eta, (\lambda_i, \mu_i, \nu_i, \xi_i, \sigma_i, \tau_i, \vartheta_i, \varphi_i, \psi_i)_{i=1}^{l_b-1}, (\omega_i, \rho_i)_{i=1}^{l_b-2},) :$

$$\begin{gathered}
c_a = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
c_d = g^{\gamma} h^{\delta} \wedge -2^{\ddot{l}} < \gamma < 2^{\ddot{l}} \wedge \\
c_m = g^{\varepsilon} h^{\zeta} \wedge -2^{\ddot{l}} < \varepsilon < 2^{\ddot{l}} \wedge \\
(\textstyle\prod_{i=0}^{l_b-1} c_{b_i}^{2^i})/c_b = h^{\eta} \wedge \\
c_{\nu_1} = g^{\lambda_1} h^{\mu_1} \wedge ... \wedge c_{\nu_{l_b-1}} = g^{\lambda_{l_b-1}} h^{\mu_{l_b-1}} \wedge \\
c_{\nu_1} = c_a^{\alpha} c_n^{\nu_1} h^{\xi_1} \wedge c_{\nu_2} = c_{\nu_1}^{\lambda_1} c_n^{\nu_2} h^{\xi_2} \wedge ... \wedge c_{\nu_{l_b-1}} = c_{\nu_{l_b-2}}^{\lambda_{l_b-2}} c_n^{\nu_{l_b-1}} h^{\xi_{l_b-1}} \wedge \\
-2^{\ddot{l}} < \lambda_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \lambda_{l_b-1} < 2^{\ddot{l}} \wedge \\
-2^{\ddot{l}} < \nu_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \nu_{l_b-1} < 2^{\ddot{l}} \wedge \\
c_{\mu_1} = g^{\omega_1} h^{\rho_1} \wedge ... \wedge c_{\mu_{l_b-2}} = g^{\omega_{l_b-2}} h^{\rho_{l_b-2}} \wedge \\
-2^{\ddot{l}} < \omega_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \omega_{l_b-2} < 2^{\ddot{l}} \wedge \\
((c_{b_0} = h^{\sigma_0} \wedge c_{\mu_0}/g = h^{\tau_0}) \vee (c_{b_0}/g = h^{\vartheta_0} \wedge c_{\mu_0}/c_a = h^{\psi_0})) \wedge \\
((c_{b_1} = h^{\sigma_1} \wedge c_{\mu_1}/c_{\mu_0} = h^{\tau_1}) \vee \\
(c_{b_1}/g = h^{\vartheta_1} \wedge c_{\mu_1} = c_{\mu_0}^{\lambda_1} c_n^{\varphi_1} h^{\psi_1} \wedge -2^{\ddot{l}} < \varphi_1 < 2^{\ddot{l}})) \wedge ... \wedge \\
((c_{b_{l_b-2}} = h^{\sigma_{l_b-2}} \wedge c_{\mu_{l_b-2}}/c_{\mu_{l_b-3}} = h^{\tau_{l_b-2}}) \vee \\
(c_{b_{l_b-2}}/g = h^{\vartheta_{l_b-2}} \wedge c_{\mu_{l_b-2}} = c_{\mu_{l_b-3}}^{\lambda_{l_b-2}} c_n^{\varphi_{l_b-2}} h^{\psi_{l_b-2}} \wedge -2^{\ddot{l}} < \varphi_{l_b-2} < 2^{\ddot{l}})) \wedge \\
((c_{b_{l_b-1}} = h^{\sigma_{l_b-1}} \wedge c_d/c_{\mu_{l_b-2}} = h^{\tau_{l_b-1}}) \vee \\
(c_{b_{l_b-1}}/g = h^{\vartheta_{l_b-1}} \wedge c_d = c_{\mu_{l_b-2}}^{\lambda_{l_b-1}} c_n^{\varphi_{l_b-1}} h^{\psi_{l_b-1}} \wedge -2^{\ddot{l}} < \varphi_{l_b-1} < 2^{\ddot{l}}))\}
\end{gathered}$$

Appendix B

The following protocol will prove that m is a prime.

1. The prover picks random $\hat{a}_i \in_R Z_m$ for $i = 1, ..., s$ and commits to them as $c_{\hat{a}_i} = g^{\hat{a}_i} h^{r_{\hat{a}}}$ with $r_{\hat{a}} \in_R Z_Q$ for $i = 1, ..., s$. She sends $c_{\hat{a}_1}, ..., c_{\hat{a}_s}$ to the verifier.

2. The verifier picks random integers $-2^l < \breve{a}_i < 2^l$ for $i = 1, ..., s$ and sends them to the prover.

3. The prover computes $a_i := \hat{a}_i + \breve{a}_i (mod\, m)$, $c_{a_i} := g^{a_i} h^{r_{a_i}}$ with $r_{a_i} \in_R Z_Q$, $d_i := a_i^{(m-1)/2} (mod\, m)$, and $c_{d_i} := g^{d_i} h^{r_{d_i}}$ with $r_{d_i} \in_R Z_Q$ for all $i = 1, ..., s$. Moreover, the prover commits to $(m-1)/2$ by $c_b := g^{(m-1)/2} h^{r_b}$ with $r_b \in_R Z_Q$. Then the prover searches a base $\tilde{a}$ such that $\tilde{a}^{(m-1)/2} \equiv -1 (mod\, m)$ holds and commits to $\tilde{a}$ by $c_{\tilde{a}} := g^{\tilde{a}} h^{r_{\tilde{a}}}$ with $r_{\tilde{a}} \in_R Z_Q$.

4. The prover sends $c_b, c_{\tilde{a}}, c_{a_1}, ..., c_{a_s}, c_{d_1}, ..., c_{d_s}$ to the verifier and then they carry out the following protocol.

$$\begin{gathered}
S_{prime} := PK\{(\alpha, \beta, \gamma, \nu, \xi, \rho, \kappa, (\delta_i, \varepsilon_i, \zeta_i, \eta_i, \vartheta_i, \omega_i, \rho_i, \kappa_i, \mu_i, \psi_i)_{i=1}^{s} : \\
c_b = g^{\alpha} h^{\beta} \wedge -2^{\ddot{l}} < \alpha < 2^{\ddot{l}} \wedge \\
c_m = g^{\nu} h^{\xi} \wedge -2^{\ddot{l}} < \nu < 2^{\ddot{l}} \wedge \\
c_b^2 g / c_n = h^{\gamma} \wedge \\
c_{\tilde{a}} = g^{\rho} h^{\kappa} \wedge (\rho^{\alpha} \equiv -1 (mod\, \nu)) \wedge \\
c_{\hat{a}_1} = g^{\delta_1} h^{\varepsilon_1} \wedge ... \wedge c_{\hat{a}_s} = g^{\delta_s} h^{\varepsilon_s} \wedge \\
c_{a_1} / g^{\breve{a}_1} = g^{\delta_1} c_n^{\zeta_1} h^{\eta_1} \wedge ... \wedge c_{a_s} / g^{\breve{a}_s} = g^{\delta_s} c_n^{\zeta_s} h^{\eta_s} \wedge \\
-2^{\ddot{l}} < \delta_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \delta_s < 2^{\ddot{l}} \wedge \\
-2^{\ddot{l}} < \zeta_1 < 2^{\ddot{l}} \wedge ... \wedge -2^{\ddot{l}} < \zeta_s < 2^{\ddot{l}} \wedge \\
c_{a_1} = g^{\rho_1} h^{\kappa_1} \wedge ... \wedge c_{a_s} = g^{\rho_s} h^{\kappa_s} \wedge \\
(c_{d_1} / g = h^{\vartheta_1} \vee c_{d_1} g = h^{\vartheta_1}) \wedge ... \wedge (c_{d_s} / g = h^{\vartheta_s} \vee c_{d_s} g = h^{\vartheta_s}) \wedge \\
c_{d_1} = g^{\mu_1} h^{\psi_1} \wedge ... \wedge c_{d_s} = g^{\mu_s} h^{\psi_s} \wedge \\
(\rho_1^{\alpha} \equiv \mu_1 \ (mod\, \nu)) \wedge ... \wedge (\rho_s^{\alpha} \equiv \mu_s \ (mod\, \nu))\}
\end{gathered}$$