社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＣＳＥＣ－26　(51)
2004／7／21

情報量的に安全な秘密多項式評価法と電子投票への応用

大塚　玲[†]　Anderson C.A. Nascimento[††]　今井　秀樹[††]

† IPA セキュリティセンター　〒113-8561 東京都文京区本駒込 2-28-8
†† 東京大学生産技術研究所　〒153-8505 東京都目黒区駒場 4-6-1
E-mail: †a-otsuka@ipa.go.jp, ††anderson@imailab.iis.u-tokyo.ac.jp, †††imai@iis.u-tokyo.ac.jp

あらまし　本論文では情報量的に安全な秘密多項式評価法 (Oblivious Polynomial Evaluation) を提案し、これに基づく電子投票方式の構成法を示す。秘密多項式評価法は、Naor と Pinkas が 99 年に提案した 2-party プロトコルであり、Alice の持つ秘密多項式と Bob の持つ秘密の値から Bob の秘密値での多項式の評価結果を Bob に出力させるプロトコルである。提案法は、攻撃者の計算能力／記憶能力などに一切の仮定を置かずに安全性を保証できる秘密多項式評価法であり、先に筆者らが与えた秘密鍵サイズの限界式を満たす最適な方式になっている。さらに、本論文ではこの秘密多項式評価法に基づいた情報量的に安全な公開検証可能秘密分散方式（Publicly Verifiable Secret Sharing: PVSS）と、これを用いた掲示板方式による電子投票方式の構成を示す。提案する電子投票方式は有権者数が 100 万人規模の電子投票においても、投票者が持つべき秘密鍵のサイズが 300MB 程度と小さく、集票作業全体の検証に要する通信量も 1000 人まで、1 万人までの結託を許す場合でそれぞれ 27GB、220GB 程度であり効率的である。
キーワード　情報量的安全性, 秘密多項式評価, 検証可能秘密分散, 電子投票

Unconditionally Secure Polynomial Evaluation And Its Application To Electronic Voting

Akira OTSUKA[†], Anderson C. A. NASCIMENTO[††], and Hideki IMAI[††]

† IT Security Center, IPA.　2-28-8 Honkomagome, Bunkyo-ku, Tokyo 113-8561, Japan
†† Institute of Industrial Science, University of Tokyo.
4-6-1, Komaba, Meguro-ku, Tokyo, 153-8505, Japan
E-mail: †a-otsuka@ipa.go.jp, ††anderson@imailab.iis.u-tokyo.ac.jp, †††imai@iis.u-tokyo.ac.jp

Abstract　In this paper, we propose an unconditionally secure oblivious polynomial evaluation (US-OPE) and a novel electronic voting scheme based on US-OPE. Oblivious Polynomial Evaluation is a 2-party protocol proposed by Naor and Pinkas in 1999 where Alice and Bob are given a secret polynomial, a secret value respectively, and after executing the protocol Bob privately outputs a value of the polynomial evaluated at the secret value without giving any useful information on their secrets. The proposed US-OPE is secure without assuming any computational/storage bound on adversary, and is an optimal construction satisfying the lower bounds previously shown by the authors. Moreover, in this paper, we will present an construction of unconditionally secure publicly verifiable secret sharing (US-PVSS) and an electronic voting scheme in the bulletin board model based on our PVSS. The proposed electronic voting scheme is efficient. For example, in a case of 1 million voters, the storage complexity to store private key required to store for each voter is as small as 300MB. Communication complexity to verify the overall tallying process is 27GB in a case of tolerating up to 1000 colluding users, and 220GB in a case of tolerating up to 10,000 users.
Key words　Unconditional Security, Oblivious Polynomial Evaluation, Verifiable Secret Sharing, Electronic Voting

1. Introduction

An invention of quantum computers [12] which efficiently solves factoring problems or discrete logarithm problems may totally break the current public-key based information systems. More practically, TWIRL [13] recently proposed by Shamir and Tromer is threatening to change our naive estimation of secure key sizes in the near future. One may hope a more secure scheme in principle which is not bothered by this everlasting game.

In this paper, we investigate several information theoretic primitives which are useful to design more complicated distributed multiparty protocols. Oblivious Polynomial Evaluation(OPE) is one of the very useful tools where two parties, Alice and Bob, are given a polynomial and a value respectively on their private inputs. Then, after jointly executing a protocol, Bob outputs a value of the polynomial evaluated at the value without learning any useful information about the values of their private inputs each other. OPE is first proposed by Naor and Pinkas [8] in the computational setting. In [2], they improved the efficiency of OPE, and also proposed OPE in the information theoretic setting. However, the security of their scheme depends on trustiness of a online trusted party. Our result shows that if private keys are securely distributed to each players, then unconditionally secure is efficiently implementable. The simplest implementation of our protocol requires a trusted party who engages only in the set-up phase (trusted initializer [10]). Thus no online trusted party is required.

The other primitive introduced in this paper is information theoretic version of Publicly Verifiable Secret Sharing (PVSS). PVSS is an extension of Verifiable Secret Sharing [3], [5] first introduced by Stadler [14] and later improved [6], [11]. PVSS is a VSS with a property that not only the players but anyone can verify that the secret is shared properly. Interestingly, PVSS is possible under the computationally bounded model or maybe under the storage bounded model, but impossible in the information theoretic model. Since the schemes based on computational assumption can utilize a public verification key to check the consistency of the shares in such a way that the adversary cannot cheat a casual user without solving a computational hard problem. However, in the information theoretic model, it is impossible since the casual user has no information theoretic advantage over the adversary. Thus, we restrict the notion of PVSS to that every user with his own private key can check the consistency of the shares even if they are not the shareholders.

This paper is organized as follows. In section 2, we will introduce a basic technique throughout this paper, hidden point evaluation. In section 3, definitions and some properties of US-OPE and US-PVSS are presented. In section 4, we will show the construction of unconditionally secure electronic voting scheme.

2. Preliminary

In this section, we first introduce the most simplified version of our information theoretic tool, hidden point evaluation technique. The setting taken in this paper is as follows. Each player is given some predistributed information described below so that each player has an information theoretic advantage over the other players. Our aim is to construct more complicated protocols like electronic voting using only this predistributed information, thus without depending on any computational assumption.

Suppose we have a prover P and verifiers $V_1, \ldots, V_n$, all of them are probabilistic polynomial-time algorithms. Let q be a prime power, and let $\mathcal{F}$ be a set of all univariate polynomials in $GF(q)[x]$.

For some polynomial $f_0 \in \mathcal{F}$ of degree ω chosen randomly and uniformly. The prover is given the polynomial f_0 as is. On the other hand, each verifier $V_1, \ldots, V_n$ is given a randomly and uniformly chosen hidden point (x_i, y_i) satisfying $y_i = f_0(x_i)$ for $i = 1, \ldots, n$ in a way that each hidden point is only known to the corresponding verifier and that the other players including the prover have no information on which point on $y = f_0(x)$ is chosen by the verifier.

Note that in Shamir's secret sharing, the dealer(prover) knowing the polynomial also knows each share delivered to each shareholder(verifier). However, in our setting, the prover knows the polynomial but has no information on the shares held by the shareholders. Thus, in our setting, each verifier (shareholder) also has an information theoretic advantage over the prover. Using this information theoretic advantage of the verifiers, we can verify the correctness of a polynomial posted by the prover.

[Definition 1] (Verification) For a polynomial $f \in \mathcal{F}$ generated by the prover, the polynomial f is called accepted by a verifier V_i, if and only if the polynomial f satisfies $y_i = f(x_i)$.

A polynomial f is evaluated by the verifier V_i whether the polynomial f passes through a hidden point (x_i, y_i). Obviously, there are many polynomials possibly accepted by the verifier. However, since the evaluation point is hidden, there exists no better way than a random guess for the prover to find a good polynomial $f \neq f_0$ to be accepted by the verifier. From this discussion, we have the following lemmas.

[Lemma 1] (Integrity) For any adversary $\mathcal{A}$ who may collapse the prover and all the verifiers except for the targeted verifier V_i, the success probability that the adversary $\mathcal{A}$ to force the verifier V_i accept a wrong polynomial $f' \neq f_0$ is exponentially small with security parameter k.

This lemma implies that once the verifier V_i accepted a polynomial $f(x) \in \mathcal{F}$, then the polynomial is *correct*, $f(x) = f_0(x)$ for all $x \in GF(p)$, with high probability. Thus, we have the following corollary.

[Corollary 1] (Unanimity) Suppose that a polynomial $f \in \mathcal{F}$ is accepted by a honest verifier V_i, then the probability that another honest verifier V_j to reject the same polynomial f is exponentially small.

Now we turn to the security for the prover. The polynomial f_0 given to the prover is unconditionally hidden from the verifiers colluding of up to ω verifiers.

[Lemma 2] (Secrecy) Suppose that the prover has a polynomial $f_0 \in \mathcal{F}$ of degree ω, and that we have any adversary $\mathcal{A}$ that have control over up to ω verifiers with hidden points $\{(x_1, y_1), \ldots, (x_\omega, y_\omega)\}$. Then, the success probability for the adversary $\mathcal{A}$ to compute $f \in \mathcal{F}$ such that f will be accepted by at least one non-colluding verifier is exponentially small.

Further, we introduce homomorphic property to the above scheme. As in many multiparty computation defined over polynomial-based secret sharing schemes, the above stated polynomial-based hidden point technique can also be equipped with homomorphic property. Observe that for any two polynomials $f_1(x)$, $f_2(x) \in \mathcal{F}$ and two points on each polynomial $(v_i, f_1(v_i))$ and $(v_i, f_2(v_i))$, any linear combination of the two polynomials $g(x) = af_1(x) + bf_2(x)$ with a, b in $GF(q)$ satisfies the following equation:

$$g(x_i) = af_1(v_i) + bf_2(v_i)$$

More general definition follows.

[Definition 2] (Homomorphism) For a polynomial $g(x) \in \mathcal{F}$ generated from a linear combination of polynomials $f_1(x), \ldots, f_n(x) \in \mathcal{F}$, and for a commitment $(a_1, \ldots, a_n)$ generated by the prover, the commitment-polynomial pair $(a_1, \ldots, a_n)$ and $g(x)$ is called `accepted` by a verifier V_i, if and only if the following equation is satisfied:

$$g(x_i) = \sum_{j=1}^{n} a_j f_j(v_i)$$

3. New Primitives

3.1 Oblivious Polynomial Evaluation

Oblivious polynomial evaluation(OPE) is an extension of the basic primitive, oblivious transfer(OT), first introduced by Naor and Pinkas [8]. OPE is a two party protocol where Alice is given a polynomial $f(x)$ on her private input, and Bob is given a value x_0 on his private input. After executing a protocol, Bob outputs a value $y_0 = f(x_0)$ (with negligible error probability) in a way that Alice has no information (or learns negligible amount of information) on the Bob's input x_0 and that Bob has no more information (or learns negligible information) on the Alice's private input $f(x)$ than that can be implied from y_0.

3.1.1 Definitions and Bounds

In [7], OPE is formalized in the information theoretic setting. We restate the definitions and bounds on US-OPE in the following.

[Definition 3] (ϵ-correct OPE) A OPE protocol π is called ϵ-`correct` if after executing the protocol π with honest players, there exists ϵ satisfying the following equation:

$$\Pr(y \neq y_0 : (\perp, y) \leftarrow \pi(f, x_0)) \leqq \epsilon$$

where y_0 is the correct output such that $y_0 = f(x_0)$.

[Definition 4] (ϵ-private OPE) Let F, X and Y be the random variables representing the polynomial f on Alice's private input, the value x_0 on Bob's private input, and y on Bob's private output. A OPE protocol π is called ϵ-`private` for Bob if for any possible behavior of Alice,

$$I(View_A; X) \leqq \epsilon$$

where $I(\cdot;\cdot)$ is Shannon's mutual information, $View_A$ is a random variable which represents Alice's view after completion of the protocol π, X is a random variable representing Bob's input x_0.

Similarly, an OPE protocol π is called ϵ-`private` for Alice if for any possible behavior of Bob, there exists ϵ such that

$$I(F; X) \leqq \epsilon,$$

$$I(F; View_B | XY) \leqq \epsilon.$$

where $View_B$ is a random variable which represents Bob's view after completion of the protocol π, Y is a random variable representing Bob's output y_0.

An OPE protocol π is said to be ϵ-`private` if it is ϵ-private for Alice and Bob. In the special case of $\epsilon = 0$, we call the protocol π is `perfectly private`.

Let K_A and K_B be random variables representing information held by Alice and Bob respectively before initiating the OPE protocol. The following theorem gives the lower bound on the initial information.

[Theorem 1] (Lower Bounds on Private Keys)

If a OPE protocol π is perfectly private, then π satisfies the following bounds.

$H(K_A) \geqq H(F)$, $H(K_B) \geqq H(X) + H(Y|X)$

Proofs are given in [7].

3.1.2 Construction

Now we will give the optimal construction of perfectly private OPE.

Protocol OPE

Initial Information: Private Keys

Alice's key: $R(x) \in GF(q)[x]$ of degree at most n,

Bob's key: (d, R_d) where $d \in GF(q)$ and $R_d = R(d)$.

OPE Phase

Alice's input: $f(x) \in GF(q)[x]$, $\deg f(x) \leqq n$,

Bob's input: $x_0 \in GF(q)$.

(1) Bob sends to Alice $e = x_0 - d$,

(2) Alice sends to Bob $g(x) = f(x + e) + R(x)$,

(3) Bob outputs $y = g(d) - R_d$.

[Theorem 2] The above stated protocol is a secure oblivious polynomial evaluation. Moreover, it is optimal regarding its private key size.

Correctness is obvious. Since if Alice and Bob are both honest, then after the completion of the above protocol, Bob outputs the correct value $f(x_0)$ with probability 1 (perfectly correct). To prove privacy for Bob, note that d is uniformly distributed and not known to Alice, thus $H(X|K_A View_A) = H(X)$ holds. Privacy for Alice follows from the fact that every action of Bob's amounts to choosing an x_0. However, given x_0 and $f(x_0)$, he can evidently simulate his view of an execution of the above protocol: he simply chooses randomly d and R_d and polynomial $g(x)$ such that $g(d) = f(x_0) + R_d$. Since this uses no further knowledge of f, the security condition $H(F|K_B View_B) \leqq H(F|XY|K_B View_B) = H(F|XY)$ holds.

Size of the private keys clearly meets the lower bound in Theorem 1 assuming uniform distribution over all inputs.

3.2 Publicly Verifiable Secret Sharing

We will introduce an information theoretic version of the powerful and important primitive, publicly verifiable secret sharing (PVSS). PVSS is first introduced by Stadler [14]. PVSS is a variant of verifiable secret sharing schemes with additional property that every casual user can verify the consistency of publicly posted encrypted shares. This property enables electronic voting schemes to enjoy the important property that every citizen can check the correctness of voting and tallying process.

Previously proposed PVSS schemes [11], [14] based on computational assumption allows any casual users can become a public verifier. The schemes uses public verification key so that every casual user can obtain the key and verify the encrypted shares posted on the bulletin board. This is a very nice feature of PVSS.

In the information theoretic PVSS (US-PVSS), verification of the shares with a single public verification key is impossible. Thus, even public verifier must be delivered a private verification key. This is because in the single public verification key setting, public verifier does not have any information theoretic advantage, thus it is always possible for the adversary (with unbounded computing power) to cheat the public verifiers with invalid shares.

3.2.1 Definitions

[Definition 5] A *US-PVSS* consists of a dealer, n particpants $P_1, \ldots, P_n$ such that each has a private encryption function E_i and a private decryption function D_i shared with the dealer, public verifiers with a private verification key, a monotone access structure $\mathcal{A} \subseteq 2^{\{1,\ldots,n\}}$, and algorithms *Share*, *Recover*, and *PubVerify* which operate as follows:

Share: The dealer uses public encryption function to distribute the shares by calculating $S_i = E_i(s_i)$ for $1 \geqq i \geqq n$. The dealer then publishes each share S_i. *Recover*: If a group of participants want to recover the secret, they run *Recover*, which has the property that $\forall A \in \mathcal{A}$ it is infeasible to calculate s from $\{S_i \mid i \in A\}$. *PubVerify* : To verify the validity of all encrypted shares, *PubVerify* is run by any inquiring party with private verification key v_k. This algorithm has the property that $\exists u\ \forall A \in 2^{\{1,\ldots,n\}}$:

$$PubVerify(v_k, \{S_i \mid i \in A\}) = 1 \Rightarrow Recover(\{D_i(S_i) \mid i \in A\}) =$$

and $u = s$ if the dealer was honest.

Note that any participants and the dealer can be a public verifier (we call them simply Verifiers in the following) if private verification key is provided.

US-PVSS must satisfy the following three properties.

Completeness: A PVSS is said to be complete if whenever the dealer is honest (and the (unique) value for s is recoverable by the participant(s)), the verifier accepts the prover's proof as valid (with overwhelming probability).

Soundness: A PVSS is said to be sound if whenever the unique s is not recoverable, the verifier accepts the prover's proof only with negligible probability.

Secrecy: A PVSS is said to be secret if any group not in the access structure can not retrieve s.

3.2.2 Construction

Our construction of US-PVSS is a combination of the hidden point evaluation technique described in the Preliminary section and the US-OPE technique introduced in the previous section. The main idea is the following. Each VSS share in our scheme is described as a polynomial. This share polynomial is a linear combination of polynomials predistributed as Dealer's VSS-key, and it is verifiable with private verification keys using hidden point evaluation technique. Furthermore, the share polynomial is encrypted using US-OPE. Thus, the original share is encrypted using the secrecy property of US-OPE. On the other hand, the encrypted share is still verifiable since US-OPE obliviously leaks one point (this

point is designed to be equal to the private verification key) on the original share polynomial.

Protocol US-PVSS

Initial Information: Private Keys

Dealer

$$\begin{cases} \text{VSS-key} & F_1, F_2 \in GF(q)[x,y] \text{ of degree } T \text{ and } t \\ \text{OPE-key} & R_j \in GF(q)[x] \text{ of degree } 2T \quad (1 \leqq j \leqq N) \end{cases}$$

Player$_j$

$$\{\ \text{OPE-key} \quad R_j \in GF(q)[x] \text{ of degree } 2T \quad (1 \leqq j \leqq N)$$

Verifier$_k$

$$\begin{cases} \text{VSS v-key} & v_k \in GF(q) \\ & F_1(v_k, y), F_2(v_k, y) \in GF(q)[y] \\ \text{OPE p-key} & v_k, R_j(v_k) \in GF(q) \quad (1 \leqq j \leqq N) \end{cases}$$

PVSS Phase

Dealer's input: secret s

Share: Dealer first chooses α depending on s such that $s = F_1(0,0) + \alpha F_2(0,0)$. A share for Player j is computed as $s_j(x) = F_1(x,j) + \alpha F_2(x,j) + R_j(x)$. Then, Dealer publishes the commitment α and a encrypted share $s_j(x)$.

Recover: Let $A \in \mathcal{A}$ be the set of players trying to recover a secret. Now they have a set of encrypted shares $\{s_j(x) \mid j \in A\}$. To recover a secret, simply compute the interpolate the secret from the decrypted shares $\{s_j(0) - R_j(0) \mid j \in A\}$.

PubVerify : Verifier$_k$ will accept (or reject) the encrypted share $s_j(x)$ with the commitment α if the following conditions satisfied:

$$s_j(x)|_{x=v_k} = F_1(v_k, y)|_{y=j} + \alpha F_2(v_k, y)|_{y=j} + R_j(v_k)$$

[Theorem 3] The above protocol is a US-PVSS satisfying *completeness*, *soundness* and *secrecy*.

Completeness is obvious. Since if the dealer is honest, all honest Verifier$_k$ accept all encrypted shares in *PubVerify* with probability 1.

To prove soundness, let A, $B \in \mathcal{A}$ be the set of players which outputs different value: $Recover(\{D_i(S_i) \mid i \in A\}) \neq Recover(\{D_i(S_i) \mid i \in B\})$. Then there exists at least 1 share S_i where $i \in A \cup B$ such that S_i is invalid, thus $S_i \neq F_1(x,i) + \alpha F_2(x,i) + R_j(x)$, and there exists at least 1 honest verifier $k \in \{1, \ldots, N\}$ who accepts the invalid encrypted share S_i. From integrity (Lemma 1) and unanimity (Corollary 1), the probability that this situation happen is less than N/q. This probability is exponentially small.

Secrecy is clear from the secrecy property of the underlying Shamir's polynomial-based secret sharing scheme and the secrecy property of US-OPE.

4. Unconditionally Secure Electronic Voting

4.1 Model

We follow the model for electronic voting as introduced by Benaloh et al. [1], [4]. The model assumes public bulletin board with which every player can post their message to it. Players are comprised of a set of tallying authorities $A_1, \ldots, A_n$, a set of voters $V_1, \ldots, V_n$, and a set of passive public verifiers. An election proceeds in two phases. The first phase is the voting phase. In this phase, each voter posts his ballot to the bulletin board. Each ballot consists of encrypted shares of his vote, its commitment to prove the consistency of the shares and a proof that the ballot contains 0 or 1 in the two-value vote. Since the voters need not be anonymous in this scheme, it is trivial to prevent double voting. Only valid ballots will be accepted. The second phase is the tallying phase. In this phase, tallying authorities are involved. They will check each ballot posted on the bulletin board. Then, they decrypt and sum up the shares, like multiparty computation, and post each sum of the shares.

The property required to voting schemes is informally stated as follows.

- Eligibility

Ensures every eligible voter posts at most one ballot.

- Privacy

Ensures the secrecy of the contents of ballots.

- Integrity

Ensures that any party, including public verifiers, can be convinced that all valid votes have been included in the final tally.

4.2 Parameters

In the following, we will use the parameters listed below.

M : number of eligible voters
m : number of participating voters ($m \leqq M$)
N : number of authorities
T : maximum number of malicious verifiers
t : maximum number of malicious authorities

4.3 Construction

A construction of electronic voting scheme based on bulletin board model [1], [4] is given in the information theoretic model. Our construction is based on US-OPE and US-PVSS described in the previous section. The construction is separated 4 phases: (1) description of private keys, (2) Voting Phase, (3) Verification Phase, (4) Tallying Phase.

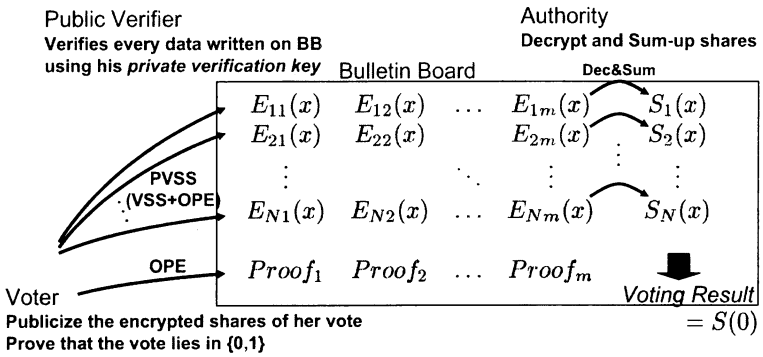

Fig. 1 Electronic Voting Scheme based on Bulletin Board Model

Protocol Electronic Voting

Initial Information: Private Keys

Voter$_i$

$$\begin{cases} \text{VSS-key} & S_{i1}, S_{i2} \in GF(q)[x,y] \text{ of degree } T \text{ and } t \\ \text{OPE-key} & R_{ij} \in GF(q)[x] \text{ of degree } 2T \quad (0 \leqq j \leqq N) \end{cases}$$

Authority$_j$

$$\{ \text{OPE-key} \quad R_{ij} \in GF(q)[x] \text{ of degree } 2T \quad (1 \leqq i \leqq M)$$

Verifier$_k$

$$\begin{cases} \text{VSS v-key} & v_k \in GF(q) \\ & S_{i1}(v_k, y), S_{i2}(v_k, y) \in GF(q)[y] \\ & \text{of degree } t \quad (1 \leqq i \leqq M) \\ \text{OPE h-key} & v_k, R_{ij}(v_k) \in GF(q) \\ & (1 \leqq i \leqq M, 0 \leqq j \leqq N) \end{cases}$$

Every player in our scheme is given a private key as Verifier: $\{\text{Verifier}\} = \{\text{Voter}\} \cup \{\text{Authority}\} \cup \{\text{Public Verifier}\}$.

Also note that OPE-key R_{ij} is shared between Voter$_i$ and Tallier$_j$ except for R_{i0}.

Voting Phase

Each participating Voter$_i$ $(i = 1, \ldots, m)$ prepares his vote as follows:

(1) Voter$_i$ decides his vote $s \in \{0,1\}$ and compute a commitment α_i satisfying: $s = S_{i1}(0,0) + \alpha_i S_{i2}(0,0)$.

(2) He computes all encrypted shares $E_{ij}(x)$ for each $j = 1, \ldots, N$ as $E_{ij}(x) = S_{i1}(x,j) + \alpha_i S_{i2}(x,j) + R_{ij}(x)$.

(3) Then, Voter$_i$ computes a proof

$$P_i(x) = f(x)(f(x) - 1) + xR_{i0}(x)$$

where $f(x) = S_{i1}(x,0) + \alpha_i S_{i2}(x,0)$.

(Note that this random polynomial $R_{i0}(x)$ is only known to the Voter$_i$, and also note that $f(0) = s$ on the second equation.)

(4) Finally, Voter$_i$ writes $i, \alpha_i, E_{i1}(x), \ldots, E_{iN}(x), P_i(x)$ on the Bulletin Board.

Verification Phase

Let $V_k(i,j,\alpha) = S_{i1}(v_k, j) + \alpha S_{i2}(v_k, j)$ be a verification function for Verifier$_k$. Everyone (say Verifier$_k$) accept (or reject) the Voter$_i$'s vote if the following conditions satisfied:

$$\begin{cases} E_{ij}(v_k) = V_k(i,j,\alpha_i) + R_{ij}(v_k) \text{ for all } j = 1, \ldots, N \\ P_i(0) = 0 \\ P_i(v_k) = V_k(i,0,\alpha_i)(V_k(i,0,\alpha_i) - 1) + R_{i0}(v_k) \end{cases}$$

Tallying Phase

Let $U_j \subseteqq \{1, \ldots, M\}$ $(j = 1, \ldots, N)$ be the set of indices which Authority$_j$ accepted in step 2. Then, Authority$_j$ $(j = 1, \ldots, N)$ sum up and decrypts all votes in U_j, and write U_j and $S_j(x)$ on Bulletin Board:

$$S_j(x) = \sum_{i \in U_j} E_{ij}(x) - \sum_{i \in U_j} R_{ij}(x).$$

Verifier$_k$ checks at least t of U_j's are equal. If so, let U be the agreed set of correct votes. Then, Verifier$_k$ accepts the output of Authority$_j$ if the following equation holds:

$$S_j(v_k) = \sum_{i \in U} V_k(i,j,\alpha_i).$$

Let $\mathcal{A}_k \subseteqq \{1, \ldots, N\}$ be the set of indices of authorities which Verifier$_k$ accepted in the previous step. Verifier$_k$ outputs the election result by reconstructing from the set of shares $\{S_i(0) \mid i \in \mathcal{A}_k\}$ if $|\mathcal{A}_k| > t$, otherwise outputs $\perp$.

4.4 Security

We briefly investigate the security of the scheme from the view points of Eligibility, Privacy and Integrity.

Eligibility is obvious, since our scheme is based on the bulletin board model where each voter is not anonymous. Thus, it is easy to resolve the situation whether a voter posted a ballot or not by checking the appropriate part of the bulletin board. Double voting is also monitored in this way.

Privacy of the content of ballots are protected by the property of US-PVSS. The information posted by each voter is in

Fig. 2 Storage Complexity

	Estimates	$M = 10^6$, T/M=1% $N = (t+1) = 10$ $q \approx 2^{80}$	$M = 10^6$, T/M=10% $N = (t+1) = 10$ $q \approx 2^{80}$
Voter	$O(NT \log q)$	313MB	342MB
Authority	$O(T^2 \log q)$ (注1)	2.3GB	200GB
Public Verifier	$O(MN \log q)$	310MB	310MB

Fig. 3 Communication Complexity

	Estimates	$M = 10^6$, m/M=10%, T/M=1%, $q \approx 2^{80}$ $N = (t+1) = 10$	$M = 10^6$, m/M=10%, T/M=0.1%, $q \approx 2^{80}$ $N = (t+1) = 3$
1 Vote	$O(NT \log q)$	220KB	80KB
Verify&Tally	$O(mNT \log q)$	220GB	8GB

the following form $(\alpha_i, E_{ij}(x), P_{ij}(x))$ where

$$E_{ij} = S_{i1}(x,j) + \alpha S_{i2}(x,j) + R_{ij}(x)$$

and

$$P_{ij}(x) = (S_{i1}(x,0) + \alpha S_{i2}(x,0))((S_{i1}(x,0) + \alpha S_{i2}(x,0) - 1) + xR_{i0}(x)$$

for $j = 1, \ldots, N$. Each verifier k has a private verification key $(v_k, R_{ij}(v_k), R_{i0}(v_k))$. Thus using this private key, the verifier can extract some information about the share $\{S_{i1}(v_k, j) + \alpha S_{i2}(v_k, j)\}$ for all $i = 1, \ldots, M$. However, any adversary who corrupts up to T verifiers cannot recover any of the share of the ballot $S_{i1}(x,j) + \alpha S_{i2}(x,j)$. Similarly, $\{P_{ij}(v_k) - v_k R_{i0}(v_k)\}$ for all k in the T-corrupting parties gives no useful information to recover $S_{i1}(x,0) + \alpha S_{i2}(x,0)$ for any $i = 1, \ldots, N$. Thus, privacy is protected.

To prove integrity, we have to show that (1) the consistency of encrypted shares, (2) security of the proof $P_i(x)$ which convinces the verifiers that the secret of each voter's PVSS lies in 0 and 1, and (3) validity of the output of each Talliers. (1) is straight forward from the property of US-PVSS. To prove (2),

It is easy to see that if the voter is honest, every honest verifier will accept the proof with the Verification algorithm described in Verification Phase of the construction.

We will consider the case that a malicious voter is trying to cheat at least one verifier, k for example, but the malicious voter can not identify who is the cheated verifier. In this case, the malicious voter is trying to post a slightly modified polynomial $P'(x)$ where $P'(x) \neq P(x)$. The verifier is trying to check the validity of the proof by checking the equation

$$\begin{cases} P'(0) = 0 \\ P'(v_k) = p_k(p_k - 1) + v_k R_{i0}(v_k) \end{cases}$$

where $p_k = S_{i1}(v_k, y)|_{y=0} + \alpha_i S_{i2}(v_k, y)|_{y=0}$. If $P'(0) \neq 0$, then every verifier reject the vote of the malicious voter. Thus, we are interested in the case that second equation eventually holds for some verifier k. Assuming the v_k and the polynomials S_{i1}, S_{i2}, R_{i0} are uniformly distributed, the probability that this case happen is $1 - (q-1)^n q^{-n} < n/q$. Thus, it is exponentially small.

Now we prove (3). In the Tallying phase, Tallier j posts $(U_j, S_j(x))$ on the bulletin board, where U_j is a set of indices of the votes which the tallier j accepted and $S_j(x)$ is a verifiable share which is a sum of every share of the votes posted for tallier j. If all the talliers are honest, then all U_j's for $j = 1, \ldots, N$ agree with the same set unless they are cheated by the voters with negligible probability (This is from unanimity property of US-PVSS). Furthermore, every honest verifier accepts the verifiable shares $\{S_j(x)\}$ and can compute the final tally. We will consider the case that there exists at least one verifier, for example k, is cheated by some malicious talliers (colluding up to t), but the malicious talliers have no idea on who is cheated. Thus, the goal of the malicious talliers is to cheat some verifier k with a wrong pair $(U'_j, S'_j(x))$ where $(U'_j, S'_j(x)) \neq (U_j, S_j(x))$ to be accepted. From the similar discussion as above, the probability that this case happen is bounded by n/q. Thus, the success probability for the malicious talliers is again exponentially small.

4.5 Efficiency

We will discuss the efficiency of the electronic voting scheme presented above. The efficiency of our scheme can be investigated in two ways: (1) Storage Complexity, (2) Communication Complexity. Since our scheme is computationally efficient. Thus we omit the evaluation of computational complexity here.

Storage complexity of our scheme is evaluated by the size of private keys required for each voter, each tallier and each verifier respectively. As described in the construction of the scheme, the storage complexity of each player is easily computed and shown in Fig. 1 with some parameters. The required storage size for the Voter is the most critical part. It requires 313MB and 342MB in the case of a million eligible voters and collusion of up to 10,000 and 100,000 players are allowed. This storage requirement includes the private keys for the public verifiers. As you can see from Fig.1, the

dominant part is the verification key for the public verifier. As for the authority, it requires 2.3GB and 200GB of storage to store the private keys respectively in these two cases. Tallying authorities are usually organizations of some representatives, thus they seem to be able to afford these storage.

Communication complexity of our scheme is shown in Fig. 2 in the case of 1 million eligible voters allowing collusion of up to 10,000 and 1,000 users. Casting one vote requires only 220KB of data to post to the bulletin board. The most heavy part is the communication for the Verify and Tally. This is because one must download the commitment posted by each voter to verify the whole tallying process.

References

[1] Josh Cohen Benaloh, Moti Yung: Distributing the Power of a Government to Enhance the Privacy of Voters (Extended Abstract). PODC 1986: 52-62

[2] Yan-Cheng Chang, Chi-Jen Lu: Oblivious Polynomial Evaluation and Oblivious Neural Learning. ASIACRYPT 2001: 369-384

[3] Benny Chor, Shafi Goldwasser, Silvio Micali, Baruch Awerbuch: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). FOCS 1985: 383-395

[4] Josh D. Cohen, Michael J. Fischer: A Robust and Verifiable Cryptographically Secure Election Scheme (Extended Abstract) FOCS 1985: 372-382

[5] Paul Feldman: A Practical Scheme for Non-interactive Verifiable Secret Sharing FOCS 1987: 427-437

[6] Eiichiro Fujisaki, Tatsuaki Okamoto: A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications. EUROCRYPT 1998: 32-46

[7] Goichiro Hanaoka, Hideki Imai, Joern Mueller-Quade, Anderson C. A. Nascimento, Akira Otsuka, Andreas Winter: Information Theoretically Secure Oblivious Polynomial Evaluation: Model, Bounds and Constructions. ACISP 2004: to appear

[8] Moni Naor, Benny Pinkas: Oblivious Transfer and Polynomial Evaluation. STOC 1999: 245-254

[9] Anderson C. A. Nascimento, Jorn Muller-Quade, Akira Otsuka, Goichiro Hanaoka, Hideki Imai: Unconditionally Non-interactive Verifiable Secret Sharing Secure against Faulty Majorities in the Commodity Based Model. ACNS 2004: 355-368

[10] R. Rivest, "Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer," manuscript.

[11] Berry Schoenmakers: A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic. CRYPTO 1999: 148-164

[12] Peter W. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5): 1484-1509 (1997)

[13] Adi Shamir, Eran Tomer: Factoring Large Number with the TWIRL Device. CRYPTO 2003: 1-26

[14] Markus Stadler: Publicly Verifiable Secret Sharing. EUROCRYPT 1996: 190-199.