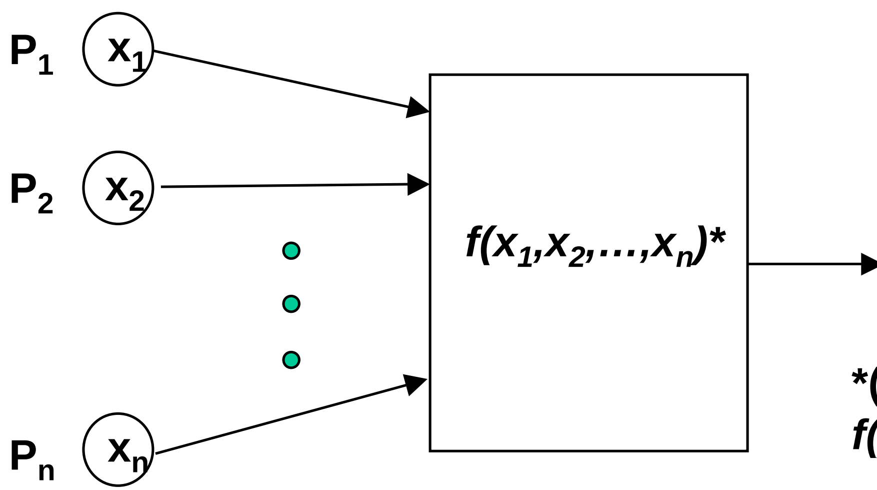
Multi-party Protocol

- (Def.) While keeping each participant's information, x_i secret, everyone can learn the result of *f()*. (If *t* malicious players exist, we say *t-secure* protocol)
-(Privacy) Even if arbitrary subset, *A* less than the half of an input set behave maliciously, any honest player except *A* can't know secret x_i of P_j.
-(Correctness) Even if *A* does any malicious acts, any P_j can know the value of *f()*.



*(Ex.)
$f() = x_1 + x_2 + \ldots + x_n$ or
$= x_1 \oplus x_2 \oplus \ldots \oplus x_n$ or
$= \max\{x_1, x_2, \ldots, x_n\}$

(w,t) Secret Sharing(I)

(Step 1) A dealer selects a secret, a_0 (< p : prime) as a constant term and $t-1$ degree random polynomial with arbitrary coefficients as :

$$h(x)=a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{t-1} \bmod p$$

(Step 2) Distributes $h(x_i)$ $(i=1,\ldots,w)$ to a share holder.

(Step 3) When t shadows $K_1, K_2, \ldots, K_t$ among w are given, recover a_0 by using the Lagrange Interpolation

$$h(x)= \sum_{s=1}^{t} K_i \prod_{j=1, j\neq s}^{t} (x - x_j)/(x_j - x_s) \bmod p$$

(w,t) Secret Sharing(II)

(Parameter) $t=3$, $w=5$, $p=17$, $a_0=13$

(Polynomial) $h(x) = (2x^2 + 10x +13) \bmod 17$

(Secret sharing) 5 shadows, $K_1=h(1)=25 \bmod 17=8$, $K_2=h(2)=7$, $K_3=h(3)=10$, $K_4=h(4)=0$, $K_5=h(5)=11$

(Recover secret) By using $K_1=8$, $K_3=10$, and $K_5=11$,

$h(x) =\{8(x-3)(x-5)/(1-3)(1-5) +10(x-1)(x-5)/(3-1)(3-5) + 11(x-1)(x-3)/(5-1)(5-3)\} \bmod 17$

$= \{8*inv(8,17)*(x-3)(x-5) + 10 *inv(-4,17)(x-1)(x-5) +11 *inv(8,17)*(x-1)(x-3)\} \bmod 17$

$=8*15(x-3)(x-5) +10*4*(x-1)(x-5)\ 11*15*(x-1)(x-3) \bmod 17$

$= 19x^2 - 92x +81 \bmod 17 = 2x^2 + 10x + 13 \bmod 17$

(Original secret) $h(0)=13$

(w,t) Secret Sharing(III)

(Parameter) t=2, w=3, a_0=011

(Polynomial) irreducible poly over GF(2^3) : p(x)=x^3+x+1=(1011)

-> f(α)=0, $\alpha^3=\alpha+1$

(Secret Sharing) h(x)=(101x + 011) mod 1011

K_1= h(001) = (101 * 001 + 011) mod 1011 = 101 +011 = 110

K_2= h(010) = (101 * 010 + 011) mod 1011 = 001 +011 = 010

K_3= h(011) = (101 * 011 + 011) mod 1011 = 100 +011 = 111

(Secret Recovering) From given K_1 and K_2,

h(x)=[110(x-010)/(001-010) +010(x-001)/(010 - 001)]mod 1011

=[110(x-010)/011 +010(x-001)/011] mod 1011

Since 011^{-1} = 110, subtraction =addition -> bit-by-bit xor

h(x) =[110*110*(x+010) +010*110*(x+001)] mod 1011

=[010 *(x+010) +111*(x+001)] mod 1011

= 010x +100 +111x +111 = 101x + 011 -> Original secret : h(0) = 011

Mental Poker

- **Non face-to-face digital poker over communication channel.**
- **No trust each other.**
- **During setting up protocol, information must be transferred unbiased and fairly. After transfer, validation must be possible.**
- **Expandability from *2* players to *n* players.**

History of Mental Poker

- SRA('79) : Using RSA
- Liption/Coppersmith('81) : Using Jacobian value
- GM('82) : Using probabilistic encryption
- Barany & Furedi ('83) : Over 3 players
- Yung('84)
- Fortune & Merrit('84) : Solve player's compromise
- Crepeau ('85) : Game without trusted dealer
- Crepaeu('86) : ZKIP without revealing strategy
- Kurosawa('90) : Using *r*-th residue cryptosystems
- Park('95) : Using fault-tolerant scheme

Basic Method

- A shuffles the card
- B selects 5 cards from A
- (Problem)
  - A can know B's selection
  - A is in advantage position than B
- (Solution)

  Use cryptographic protocols

Mental Poker by RSA

(Preparation) A and B prepare public keys (E_A, E_B) and secret keys (D_A,D_B) of RSA cryptosystem.

(Step 1) Using his public key, B posts all 52 encrypted cards ($E_B(m_i)$) in the deck.

(Step 2) A selects 5 cards in the deck and sends them to B. B decrypts ($D_A(E_A(m_i))=m_i$) using his secret key and keep them as his own cards.

(step 3) A selects 5 cards from the remaining 47 cards and encrypts using his public key ($E_A(E_B(m_j))$) and sends them to B.

(step 4) B decrypt 5 cards using his secret key and send ($E_A(m_j)$) to A

(step 5) Using his secret key, A decrypts $E_A(m_j)$ and keeps them as his cards.

(Victory or defeat) Reveal his own cards to counterparts and decides

(Validation) Reveal his secret card to counterpart

References

- A. Shamir, R.L.Rivest,L.M.Adleman, "Mental Poker", MIT Technical Report, 1978
- M.Blum, "Mental Poker", 1982
- S.Goldwasser, S.Micali, "Probabilistic Encryption & How to play mental poker keeping secret all partial information", Proc. of 14th ACM STOC Meeting, pp.365-377,1982,
- I.Barany, Z.Furedi, "Mental Poker with three or more players", 1983
- O.Goldreich, S.Micali, A. Widgeson, "How to play any mental game or a completeness theorem for protocols for honest majority", Proc. of STOC, 1987
- A.Wigderson, "How to play any mental game or a completeness theorem for fault-tolerant distributed protocols", Former(?) version of GMW paper, 1987
- K.Kurosawa, Y.Katayama, W.Ogata, S.Tsujii, "General public key residue cryptosystems and mental poker protocols", Proc. of Eurocrypt'90, pp.374-368, 1990

Electronic Vote

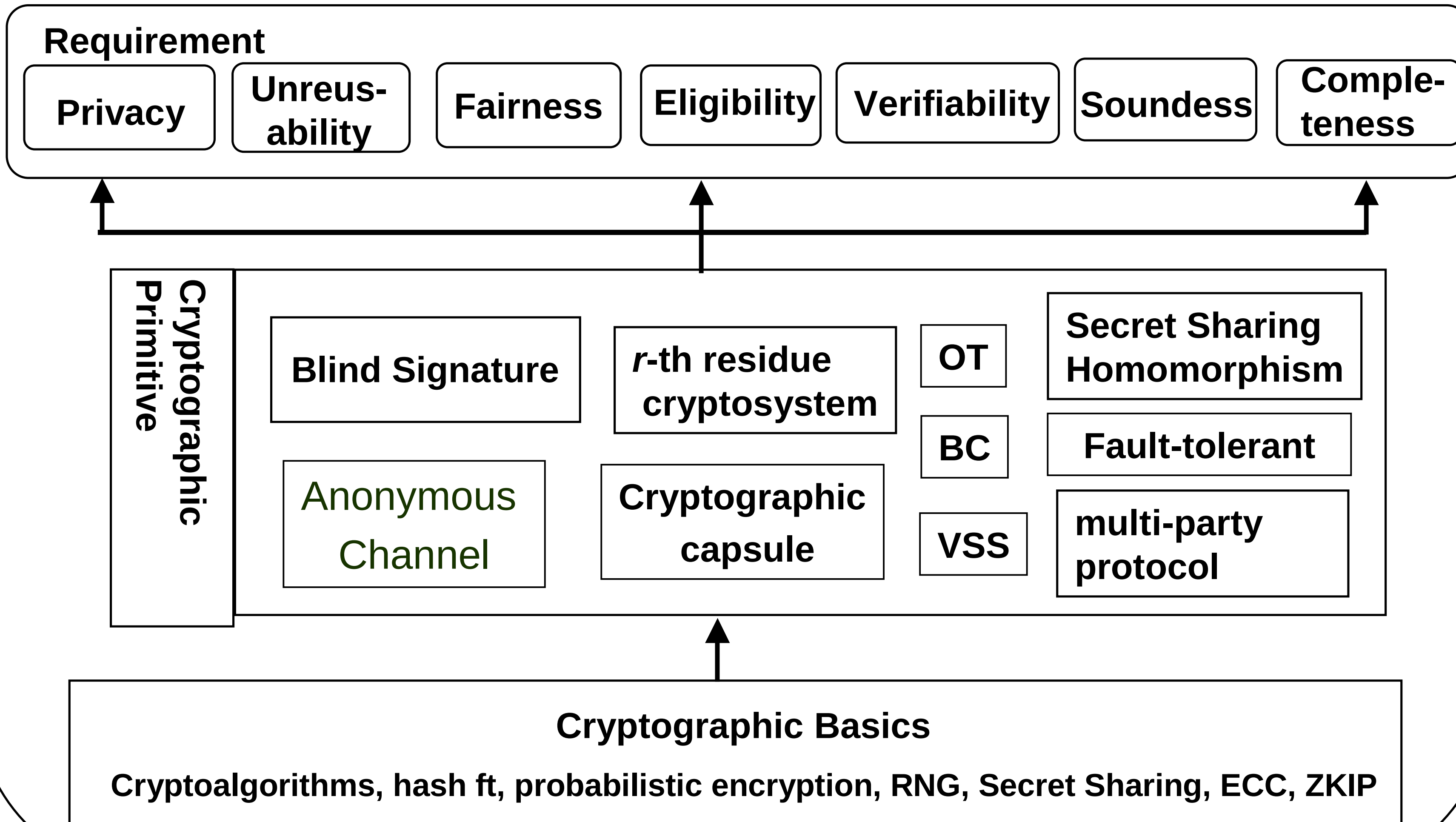
❑ Yes-No Vote

- While keeping each voter's vote secret *(x_i)*, compute only total sum ($T=x_1+x_2+\ldots+x_n$)
- Malicious players among n exist (interruption etc.)
- t-secure multiparty protocol
- Basic tool
  - VSS (Verifiable Secret Sharing)
  - OT (Oblivious Transfer)

Requirement of E-vote

- **Privacy : keeping each vote secret**
- **Unreusability : prevent double voting**
- **Fairness : if interruption occurs during voting process, it doesn't affect remaining voting**
- **Eligibility : only eligible voter can vote**
- **Verifiability : can't modify voting result**
- **Soundness : preventing malicious acts**
- **Completeness : exact computation**

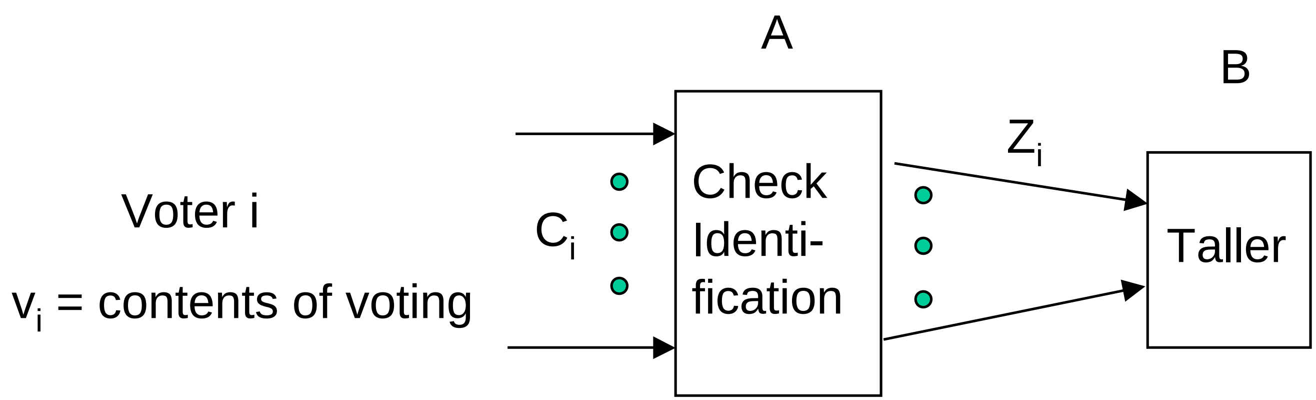
Cryptographic tool for e-vote



Implementation Methods

- Using RSA
  - Koyama (NTT), Meritt(America) etc
  - Assuming trustful center
- Using r-th residue cryptosystem
  - Small-scale vote by Kurosawa(TIT)
- Application of multiparty protocol
  - Benaloh(America), Iverson(Norway) etc
  - Keeping voter's vote secret, small-scale yes-no vote
- Using Anonymous Channel
  - Chaum(Netherland), Ohta/Fujioka(NTT), Sako(NEC), Park(Korea) etc
  - Unlinking vote and voting, suitable for large scale voting
- Others
  - Using Non-anonymous channel (Okamoto)
  - multi-recastable ticket
  - receipt-freeness: prevent buying vote, coercion

E-voting using RSA



(Procedure of Voting)

(Step 1) A voter votes by computing $C_i = E_A(D_i(E_B(v_i)))$

(Step 2) After checking identification, Center A sends $Z_i = E_B(D_A(E_i(D_A(C_i))) = E_B(D_A(v_i))$ to B.

(Step 3) B make $v_i = D_B(E_A(Z_i))$ to be public.

* $v_i = D_B(E_A(E_B(D_A(E_i(D_A(E_A(D_i(E_B(v_i)))))))))$ -> reblocking problem

E-voting by PKC

- A voter sends his vote by encrypting center's public key.
- Center decrypts each votes by its secret key and accumulate each vote.
- (Problem)
  - Reveal of privacy
  - Center's malicious acts : post it in the bulletin board

r-th residue

(Def.) Given integer *n*, an integer *z* is called as *r*-th residue mod. *n* iff $\exists$ some integers *x* s.t. $z = x^r$ mod n.

(Notation) Z_n^r : set of *r*-th residues mod. *n* which are relatively prime to *n*, $\_Z_n^r$: set of *z* in Z_n^* which are not *r*-th residues mod *n*.

(Lemma)

1. Z_n^r is a subgroup of Z_n^*

2. Given a fixed *r* and *n*, every integer *z* in Z_n^r has the same number of *r*-th roots.

3. If r and $\varphi(n)$ are relatively prime, every integer *z* in Z_n^* is an *r*-th residue mod n (*i.e.*, $Z_n^r = Z_n^*$) and r-th root of *z* is given by z^A mod *n* where *A* satisfying $Ar-B\varphi(n)=1$.

r-th residue cryptosystem(I)

- secret key : primes p,q
- public key : N (= pq), y
- message : m ($0 \le m < r$), r(*): random number
- encryption [KKOT90]
  - $E(m) = y^m x^r$ mod N (x : random number)
  - $E(m) \bullet E(n) = y^m x_1^r \bullet y^n x_2^r$ mod N

    $= y^{(m+n)} (x_1 x_2)^r$ mod $N = y^{(m+n)} z^r$ mod N

  Thus, $E(m+n)=E(m)E(n)z^r$ mod N for some z

  (additive homomorphism)

  (*) If $r=2$[GM82], $(y/p)=(y/q)=-1$.

  prime r [CF85][BY85], $r \mid p-1$, $r \nmid q-1$, y is r-th non-residue.

r-th residue cryptosystem(II)

Decryption

- $y^j \notin B_N(r),\ 1 \le j < r,\ B_N(r) = \{w | w = x^r \bmod N,\ x \in Z_N^*\}$
  - $gcd(p-1,r)=e_1,\ gcd(q-1,r)=e_2$
  - $r=e_1e_2$ if r is odd, $r=(e_1e_2)/2$ if even
  - $gcd(e_1,e_2)$ is 1 if r is odd, 2 if even
  - $(y/N)=1$ if r is even.
- Under mod p $\{E(m)\}^{(p-1)/e_1}=(y^m x^r)y^{(p-1)/e_1}=(y^{(p-1)/e_1})^m(x^{r/e1})^{(p-1)}=(y^{(p-1)/e_1})^m$
- Similarly under mod q, $\{E(m)\}^{(q-1)/e_2}=(y^{(q-1)/e_2})^m$
- Thus, for $0 \le i < r$, compare $\{E(m)\}^{(p-1)/e_1}$ and $\{E(m)\}^{(q-1)/e_2}$ with $(y^{(p-1)/e_1})^i$ and $(y^{(q-1)/e_2})^i$ respectively

E-voting(1) – 1 center -

- **Basic Protocols**
  - **(1) Center publishes *r*-th residue cyptosystem's public key (*N,y*). (# of voters, h are less than *r*)**
  - **(2) Each voter i encrypts his vote depending on m_i=0 or 1 and sends $E(m_i)=y^{m_i} x_i^r \bmod N$ to a center (x_i is a large random number.)**
  - **(3)Center publish $M = m_1 + m_2 + \ldots + m_h$ to the public**

E-voting(2) - 1 center -

(1) Center shows that "(N,y) is public key information of r-th residue cryptostem in ZKIP"

(2) Each voters show that "The plaintext of $E(m_i)$ is $m_i=0$ or 1 in ZKIP" (cryptographic capsule)

(3) Center shows that "In order that $E(m_1) \cdots E(m_h) = y^M x^r \bmod N$ (where $M=m_1 + \ldots + m_h$), prove that $z=y^M x^r \bmod N$ ($x=x_1 \ldots x_h$) in ZKIP.

Problems

- Center can know each voter's vote
- Multiple centers
  - center 1 : N_1, y_1
  - ...
  - center n : N_n, y_n

Multiple centers

- Voter *i*
  - $m_i = m_{i1} + \ldots + m_{in} \bmod r$
  - $E(m_{i1})$ -> center 1 , …
  - $E(m_{in})$ -> center n
- Center *j*
  - $E_j(M_{1j})$
  - $E_j(M_{2j})$
  - ...
  - $E_j(M_{kj})$

  Publish $M_j = M_{1j} + \ldots + M_{kj}$
- Voting result
  - $M = M_1 + \ldots + M_n$

Problems of multiple centers

- If a center fail, voting fails too.
  → Introducing Secret Sharing Scheme.

- If a voter can play as a center, we don't need a center.

E-voting using SSS

- **Voter *i***
  - $f_i(x) = m_i + a_1x + \ldots + a_{k-1}x^{k-1}$
  - $E_1(f_i(1))$: to center 1, $E_2(f_i(2))$: to center 2, …, $En(f_i(n))$: to center n
  - If only *k* centers cooperate, we can know m_i.
- **Center j publishes $M_j = f_1(j) + \ldots + f_n(j)$**
  - $f(x) = f_1(x) + \ldots + f_n(x)$

    $=(m_1 + \ldots + m_k) + a'_1x + \ldots a'_{k-1}x^{k-1}$

    , $f(j) = M_j$
  - Even if *(n-k)* centers fail, if we know *k* M_j's

    then recover $(m_1 + \ldots + m_k)$.

Verification

- **Voter i**

  $$f_i(x) = m_i + a_1 x + \ldots + a_{k-1} x^{k-1}$$

  $$\begin{cases} y_1 = E_1(f_i(1)) & \text{: to center 1} \\ \ldots & \\ y_n = E_n(f_i(n)) & \text{: to center n} \end{cases}$$

- **To show that $(y_1, \ldots, y_n)$ is computed by above equations in ZKIP -> VSS (Benaloh'86)**

Theorem from ZKIP

- **If there is a <u>secure probabilistic encryption</u>, then <u>every language in NP has ZKIP</u> in which the prover is a probabilistic polynomial-time machine that gets an NP proof as an auxiliary input [GMW85] .**
- **An encryption system secure as in [GM84] is a probabilistic poly-time algorithm *f* that on input *x* and internal coin tosses *r*, outputs an encryption *f(x,r)*. Decryption is unique : that is *f(x,r) = f(y,s)* implies *x=y*.**

VSS(I)

SS+ZKP

(Purpose) To show a dealer behaves in a right way, (i.e. any number of more than *k* shareholders can reveal same secret in ZKIP).

(1) A dealer encrypt a secret, *m* to *c(m)* and send it to n shareholders.

(2) Using SSS, a dealer sends *f(j) (j=1,…,n)* to each shareholder *j*.

(3) A dealer show each shadows was constructed by the above procedure by using ZKIP

(Tools) Checking each shadow in a correct way is NP problem. If there is 1-way function, there always exist ZKPS to prove this.

VSS(II)

- (Assumption) arbitrary 1-way permutation
- (*k,n*) secret $s \in Z_p$
- [Preparation] Sender *k-1* degree random polynomial over Z_p^* and computes n shares.
- Senders encrypt *i*-th piece with user *i*'s PKC.
- Sender provide each receiver with ZKP that encrypted messages correspond to the evaluation of a single polynomial over Z_p^* and applying *f* to the constant term of this polynomial yield *s*.

VSS using *r*-th residue cryptosystem(I)

(step1) A dealer encrypts the i-th shareholder's secret, $s_i = f(i)$ by using r-th residue cryptosystem, $z_i = y_i^{s_i} x_i^r \bmod N_i$ and makes it public. The i-th shareholder decrypts this and recover his secret information, s_i.

The following is considered as ZKIP about

$L = \{z_1, \ldots, z_n \mid z_i = y_i^{s_i} x_i^r \bmod N_i, s_i = f(i)\}$. Repeat steps (2)~ (4) t times, t= number of bits in N.

(step2) A dealer selects random polynomial f' of degree (k-1) and computes the same as (step 1). i.e., a dealer

computes the i-th shareholder's secret, $s'_i = f'(i)$ by using r-th residue cryptosystem, $z'_i = y_i^{s'_i} x'^r_i \bmod N_i$. The i-th share holder decrypts this and recovers his secret information s'_i .

VSS using *r*-th residue cryptosystem(II)

(step 3) The shareholders send *e*=1 or 0 to a dealer. (All shareholders agree the value of *e*).

(step 4) If *e*=0, the dealer reveals all s'_i and x'_i and shows *f'* has degree of *(k-1).* If *e=1*, the dealer shows all t_i and w_i satisfying $z_i z'_i = y_i^{t_i} w_i^r \bmod N_i$ and *f+f'* has degree of (*k-1*).

(Example) A voter sends his vote to *n* centers, it is hard to reveal his secret voting without collaborating more than *k* centers.

OT(Oblivious Transfer)(I)

(Purpose) While keeping secret, sending the corresponding information.

(Ex) OT : Alice has a secret bit, b. At the end of protocol, one of the following two events occurs, each with probability 1/2.

(1) Bob learns the value of *b*.

(2) Alice gains no further information about the value of *b*

(other than what Bob knew before the protocol)

[Result] If there exists PKC, feasible to construct OT[EGL85]

[Application] electronic contract signing, multi-party protocol, etc.

OT(Oblivious Transfer)(II)

P_A : A's public key
S_A : A's secret key
$D_{SA}(E_{PA}(x))=x$
$E_{PA}(x) : Z_n \to Z_N$
- If RSA, $E(n,e) : Z_n \to Z_n$
- If DH, $E(p,g) : Z_{p-1} \to Z_p \times Z_{p-1}$

A | **B**

A: $r_0, r_1 \in_U Z_N$

A → B: P_A, r_0, r_1

B: $b \in_U \{0,1\}$, $x \in_U Z_n$
$q=E_{PA}(x) + r_b \bmod N$

B → A: q

A: $y_i = D_{SA}(q - r_i \bmod N)$
where $i=0, 1$
$c_0 = m_0 + y_0 \bmod n$
$c_1 = m_1 + y_1 \bmod n$

A → B: (c_0, c_1)

B: $m_b = c_b - x \bmod n$

B can derive m_b,but can't derive $m_{b \oplus 1}$ because it is equivalent to derive $D_{SA}(E_{PA}(x)+r_b - r_{b \oplus 1} \bmod N)$ which is hard to solve PKC itself.

OT (III)

[1-2 Oblivious String Transfer]

Alice has 2 strings, S_0 and S_1. Bob has a selection bit, *s*. At the end of protocol, the following three conditions hold.

(1) Bob learns the value of S_s.

(2) Bob gains no further information about the value of S_{1-s}.

(3) Alice learns nothing about the value of *s*.

Alice has 2 secret strings. Bob select exactly one of them, and Alice doesn't know which secret Bob selected.

[Oblivious Circuit Evaluation] Alice has some secret, i, and Bob has some secret, j. Both agreed on some circuit f. At the end of protocol, the following three conditions holds.

(1) Bob learns the value of f(i,j).

(2) Bob learns no further information about j (other than that revealed by knowing i, f(i,j).

(3) Alice learn nothing about i or f(i,j).

Anonymous Channel(I)

(Def 1) A channel is a set of probabilistic polynomial time Turing machines ($P_1,\ldots,P_n,S_1,\ldots,S_n$) together with a public board. P_i is called a sender, S_i is called a shuffle machine agent. P_i or S_i is called a player.

(Def 2) Let m_i be input of P_i and OUT=$\{o_1,\ldots,o_n\}$ be the final list of public board.A channel is called an anonymous channel if the following conditions hold.

[Completeness] If every player is honest, $\{o_1, \ldots,o_n\}=\{m_1,\ldots,m_n\}$.

[Privacy] For any i, the correspondence between P_i and m_i is kept secret.

An election scheme is an anonymous channel with the following condition.

[Verifiability] If $\{o_1,\ldots,o_n\} \neq \{m_1,\ldots,m_n\}$, every P_i can detect this fact with overwhelming probability.

Anonymous Channel(II)

Simple Mix Anonymous Channel

(Preparation)Sender : A_1,…A_n, Receiver: B_i , B_i's public key : E_{Bi}, Role of shuffle agent S_i : decrypting each sender's encryption, removing its random part, and sorting alphabetical order then output S_i's public key :E_i

(Purpose) Each sender doesn't know the corresponding information of message, m_i.

(step 1) Each A_i chooses a random number R and writes C_i= $E_1(R \circ B_i \circ E_{Bi}(m_i))$ on the public board.

(step 2) S_1 decrypts and throws away R, and then writes $\{B_i \circ E_{Bi}(m_i)\}$ on the public board in lexicographical order.

This gives that everyone except S_1 can't tell the correspondence between $\{A_i\}$ and $\{B_i\}$.

If a Mix is dishonest, it will be big problem.!

E-vote by anonymous channel(I)

(To prevent malicious acts of Mix)

[Registration phase]

(step 1) Each P_i chooses (K_i, K_i^{-1}) where K_i is public key and K_i^{-1} is its secret key. P_i writes $E_1(R_1 \circ E_2 (R_2 \ldots E_k(R_k \circ K_i) \ldots))$ on the public board with his digital signature.

(step 2) The k MIXes anonymous channel shuffles $\{K_i\}$ in secret.

(step 3) S_k writes K_i on the public board in lexicographical order.

Let the list be (K'_1, K'_2,…).

[Claiming phase]

(step 4) Each P_i checks that his K_i exists in the list. If not, P_i objects and election stops. If no objects in some period of time, goto the next phase.

E-voting by anonymous channel(II)

[Voting phase]

(step 5) Each P_i writes $E_1(R_1 \circ E_2(R_2 \ldots E_k(R_k \circ (K_i \circ K_i^{-1} (V_i \circ 0^l)))\ldots))$ on the public board with his digital signature.

(step 6) After the voting is over, the k MIXes anonymous channel shuffles $K_i \circ K_i^{-1}(V_i \circ 0^l)$ in secret.

(step 7) S_k writes $K_i \circ K_i^{-1}(V_i \circ 0^l)$ on the public board in lexicographical order. Let the list be $(u_1 \circ v_1)$, $(u_2 \circ v_2)$,…

(step 8) Everyone checks that $u_i = K'_i$ and $u_i(v_i) = * \ldots * 0^l$ for each i. If the checks fails, stop.

(step 9) It is easy for everyone to obtain $\{V_1,\ldots,V_n\}$.

Other e-voting scheme

- McEliece (‘81) : (k,n) threshold scheme : BCH codes
- Ben-Or, Goldwasser, Widgerson(‘88) : using error-correcting capability of BCH to tell a center lying

References(I)

- J.C.Benaloh, “Secret sharing homomorphisms : keeping shares of a secret”, Crypto’86, pp.251-260, 1986
- D.Chaum, “Untraceable electronic mail, return addresses and digital pseudonyms”, Com. Of ACM, 24,2, pp.84-88, 1981
- D.Chaum, “Elections with unconditionally-secret ballots and disruption to breaking RSA”, Proc. of Eurocrypt’88, pp.177-182,1988
- J.Cohen and M. Fischer, “A robust and verifiable cryptographically secure election scheme”, Proc. of 26th IEEE symp. On FOCS, pp.372-382, 1985
- S.Even, O.Goldreich and A. Lempel, “A randomized protocol for signing contracts”, Com. Of ACM, 28, 6, pp.637-647, 1985
- A.Fujioka, T.Okamoto and K. Ohta, “A practical secret voting scheme for large scale election”, Proc. of Auscrypt’92, 1992
- K.Iverson, “A cryptographic scheme for comoputerized general elections”, Advances in Cryptology, Proc. of Crypto’91, pp.405-419, 1992
- Choonsik Park,et al, “실용적 선거 방식과 익명 통신로”, SCIS93-6C, 1993
- Koyama Kenji, “RSA 공개키 암호를 이용한 안전한 무기명 투표 방식”, Trans. of IEICE, J68-D, 11, pp.1956-1966, 1985
- H.Nurmi, A.Salomaa and L. Santean, “Secret ballot elections in computer networks”, Computer & Security, 10,6, pp.553-560, 1991
- T.Okamoto, A.Fujjoka and K.Ohta, “A practical large scale secret voting scheme based on non-anonymous channels”, SCIS93-1C, 1993

References(II)

- K.Sako, "Electronic voting system with objection to the center", SCIS92-13C, 1992
- K.Sako, "Electronic voting system allowing open objection to the tally", SCIS93-1B,1993
- J.C.Benaloh and M.Yung, "Distributing the power of a government to enhance the privacy of voters". Proc. of 5th ACM Symp. on Principles in Distributed computing, pp.53-62,1986
- J.C.Benaloh and D.Tuinstra, "Receipt-free secret ballot elections", Proc. of 26th ACM STOC, pp.544-553, 1994
- K.Sako and J.Killian, "Secure voting using partially compatible homomorphisms", Proc. of Crypto'94, pp.411-424, 1994
- K.Sako and J.Killian, "Receipt-free Mix type voting scheme - a practical solution to the implementation of a voting booth", Proc. of Eurocrypt'95, pp.393-403, 1995
- C.Boyd, "A New Multiple Key Cipher and an Improved Voting Scheme", Proc. of Eurocrypt'89, pp.617-625, 1990
- C.S.Park, K.Itoh and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme", Proc. of Eurocrypt'93, pp.248-259, 1993
- K.Kurosawa, Y.Katayama, Y.Ogata and S.Tsujii, "General public key residue cryptosystems and mental poker protocols", Proc. of Eurocrypt'90, pp.374-388, 1990
- A.Pfitzmann and M.Waider, "Networks without user observability -design options", Proc. of Eurocrypt'85, pp.245-253, 1986
- D.L.Chaum, "The dining cryptographers problem : unconditional sender and receipt untraceability", J. of Cryptology, Vol.1, No.1, pp.65-75, 1988
- B.Phitzmann and A.Pfitzmann, "How to break the direct RSA implementation of MIXes", Proc. of Eurocrypt'89, pp.373-381, 1989

References(III)

- C.Rackoff and D.R.Simon, “Cryptographic defense against traffic analysis”, Proc. of 25th ACM STOC, pp.672-681, 1993
- M.Waider and B.Pfitzmann, “The dining cryptographers in the disco : unconditional sender and recipient untraceability with computationally secure serviceability”, Proc. of Eurocrypt’89, pp.690, 1990
- M. Waider, “Unconditional sender and receipt untraceability in spite of active attacks”, Proc. of Eurocrypt’89, pp.302-319, 1990
- J.Bos and B. den Boer, “Detection of disrupters in the DC protocol’, Proc. of Eurocrypt’89, pp.320-327, 1990
- B.Beaver and D.Goldwasser, “Multiparty computations with faulty majority”, Proc. of 30th annaul IEEE Symp. On FOCS, pp.468-473, 1989
- O.Goldreich, S.Micali and A.Wigderson, “How to play any mental game”, Proc. of 19th ACM STOC, pp.218-229, 1987
- V.Niemi and A.Renvall, “How to prevent buying of voters in computer elections”, Proc. of Asiacrypt94, pp.164-170, 1995
- Choonsik Park, “A Study on Security and Efficiency of Cryptographic Protocols”, Ph.D Dissertation, Tokyo Institute of Tech., 1995
- B. Chor, S.Goldwasser, S.Micali, B.Awerbach, “Verfifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults”, Proc. of FOCS, pp.383-395, 1885