Secret Sharing and Threshold Cryptography

密碼學與應用

海洋大學資訊工程系

丁培毅

Introduction

- Story #1: A millionaire put all his estate in a safe and leaves the combination to his seven children. He wants it to be fair such that no single children can get the money without the cooperation of all others.
- Story #2: In the pentagon, two out of three generals have to turn the keys at the same time to launch a nuclear missile.
- Story #3: Two bank managers keep a pair of keys to the bank vault. Two of them have to come together to open the vault.
- Story #4: Documents announced by a government office may require joint signature of some officials.
- Story #5: Some company may require two employees inspect together important encrypted mails.

Introduction

- Story #6: In a certification authority (CA) system, the security of cryptographic keys is a major system design issue. It's better that several people share the cryptographic keys either to issue a certificate or to access the archive of all certificates.
- Story #7: Multiparty computation: A group of people get together and compute any function of many variables. Each participant provides one or more variables. The result is known to someone (or anyone) but no one learns anything about the inputs of other members except what is obvious from the output.
  - calculate average salary without letting others know your salary
  - comparing who is older / comparing whose bid is higher
  - two people can determine whether they share the same fetish
  - electronic voting

(information theoretic MPC)

Passive vs. Active Adversaries

- Passive adversary: a person who obeys the protocol but might either leak the secret or probe something prohibited
- Active adversary: a person who might not only leak the secret but also disrupt the protocol

Goals of Threshold Protocols

- Two divergent goals:
  - data secrecy: it's too dangerous to trust a single person
    Why not separate the secret into n disjoint shares and distributed to n people?
    Fragile integrity control: if any one person refuses to provide the share for the recovery of original secret.
  - data integrity / availability: it's too dangerous to keep only a single copy of a piece of important data
    Why not duplicate the data into n copies, so that the loss of up to n-1 copies of data is still tolerable?
    Fragile secrecy control: any one out of these n copies can leak to an adversarial party.

Active adversary

Goals of Threshold Protocols

- (t, n) threshold protocol:
  - $t \leq n$, t is the threshold, n is the number of players
  - maintain *secrecy* in the presence of up to any t-1 adversaries
  - achieve *data integrity and availability* with the cooperation of any t shareholders

Both requirements are satisfied partially.

- Assumptions: To use a (t, n) scheme, we assume implicitly
  - In case of passive adv.: # adv. ≤ t-1
  - In case of active adv.: # adv. ≤ t-1 and # adv. ≤ n-t
    (# adv. ≤ min(t-1, n-t) < n/2)

Combinatorial Secret Sharing

- Problem: Thirteen scientists are working on a secret project. They wish to lock the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. (6, 13)

  If only traditional pad locks are available

  What is the smallest number of locks needed?

  What is the smallest number of keys each scientist must carry?

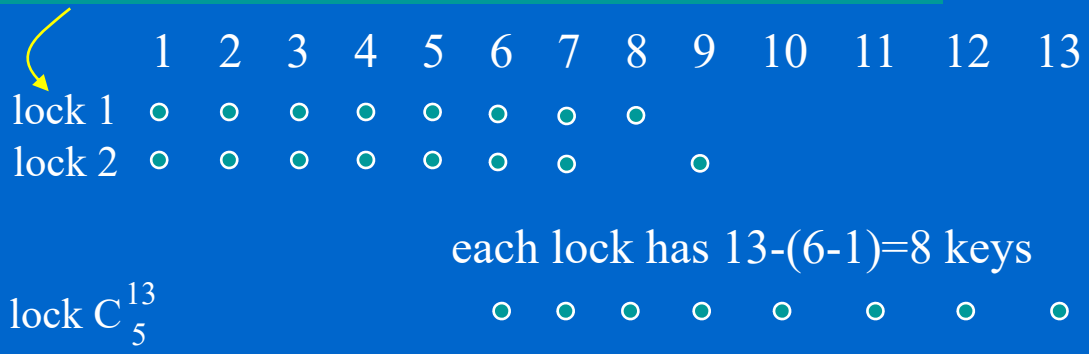
assumptions:

1. the cabinet can be locked by as many locks as you wish
2. each key can be copied as many times as you wish
3. each lock can be opened using one matched key

idea:"prevent any 6-1=5 scientists to open the cabinet"

Combinatorial Secret Sharing

Prevent {9, 10, 11, 12, 13} to open the cabinet



At least C_5^{13} locks.

C_5^{13} (13-5)/13 keys/person.

solution:

1. each lock has exactly 13-(6-1)=8 keys (minimal keys)
2. for any 6-1=5 scientists, there is exactly one lock that can not be opened (minimal locks)

note: 1. If # keys/lock > 8, this lock only locks group of 4 or less people group, this lock is not in its full power.

2. If # keys < 8, this lock locks some 6-people groups. The requirement is not satisfied.

Algebraic Secret Splitting

- Additive secret splitting

  $s = s_1 + s_2 + \ldots + s_n$

  $x, s, a_i, x_i, s_i \in Z_p$

- Multiplicative secret splitting

  $s = s_1 \, s_2 \ldots s_n$

- Polynomial secret splitting

  $f(x) = s + a_1 x^1 + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$

  $s_1 = f(x_1), \; s_2 = f(x_2), \ldots \; s_n = f(x_n)$

- In the above schemes, $\{s_i\}$ are distributed to n players
- knowing any partial set of s_j are not sufficient to recover the secret s

Properties of Secret Sharing

- No partial information of the secret can be deduced from any subset of shares.
- No assumption on the computation power of adversaries. The probability of an unexposed secret $\Pr\{s = a\} = 1/p$
- Once the secret is reconstructed, it is exposed and all shares become useless --- one-time secret sharing.
- For joint signature applications: require additional mechanism to reuse the shares --- function sharing.

Two basic models of threshold cryptography.

Properties of Secret Sharing

- In some protocols, a trusted person (the dealer) is assumed to do the sharing. In some other protocols, the secret is determined collectively by shareholders who choose their individual shares without knowing other's shares.
- Basic secret splitting scheme can be modified to a (t, n) threshold scheme in which t out of the n shares are required to reconstruct the secret s.

Shamir's Secret Sharing

- 1979, "How to share a secret," Comm. ACM 1979
- basic ideas: two points are required to determine a line; three points are required to determine a quadratic curve
- (t, n) threshold scheme: choose a prime p, $p > n$, $p > s$, s is the secret to be shared, n is the number of participants, all computations is carried out mod p, choose randomly a_1, $a_2, \ldots a_{t-1}$

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}$$

$$s_1 = f(x_1), \; s_2 = f(x_2), \ldots \; s_n = f(x_n)$$

$\{x_i\}$ are distinct public ID's for each participants, $\{s_i\}$ are their secret shares

Reconstruction of Secret

m out of *n* shareholders ($m \geq t$) get together and provide their shares $\{(x_i, s_i)\}$, they want to recover the secret *s*.

- linear system approach

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^{t-1} \\ 1 & x_2 & \cdots & x_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{t-1} \end{pmatrix} \begin{pmatrix} s \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix} \pmod{p}$$

- For m=t, the matrix is known as a Vandermonde matrix. The determinant of this matrix is nonzero, which guarantees that the linear system has a unique solution.

$$\det V = \prod_{1 \leq j < k \leq m} (x_k - x_j)$$

- For $m>t$, the rank of this matrix is only *t* (there is only *t* independent equations, the others are just dependent ones). Take an arbitrary *t* subsets to reconstruct the secret *s*.

Lagrange Interpolation Polynomial

- let I be the set of shareholders who want to participate in reconstruction, $|\mathrm{I}| \geq t$

$$p(x) \equiv \sum_{k \in \mathrm{I}} y_k \, l_k(x) \pmod{p}$$

$$l_k(x) \equiv \prod_{\substack{i \in \mathrm{I} \\ i \neq k}} \frac{x - x_i}{x_k - x_i} \pmod{p} \quad \text{such that} \quad l_k(x_j) \equiv \begin{cases} 1 & when \ j = k \\ 0 & when \ j \neq k \end{cases} \qquad j \in \mathrm{I}$$

- the reconstructed secret is

$$p(0) \equiv \sum_{k \in \mathrm{I}} y_k \prod_{\substack{i \in \mathrm{I} \\ i \neq k}} \frac{-x_i}{x_k - x_i} \pmod{p}$$

Example: (3,8) - Threshold Scheme

- Sharing Phase: trusted dealer prepares
  - secret s = 190503180520 “secret”
  - choose randomly a prime p = 1234567890133 > s
  - degree two polynomial f(x) = $s + a_1 x + a_2 x^2$:
    choose randomly a_1= 482943028839, a_2= 1206749628665
  - eight shares:

(1, 645627947891)	(5, 675193897882)
(2, 1045116192326)	(6, 852136050573)
(3, 1544000023692)	(7, 973441680328)
(4, 442615222255)	(8, 1039110787147)

(3,8) - Threshold Scheme (cont'd)

- Reconstruction phase: any 3 or more shareholders
  - ex. (2, 1045116192326) (3, 154400023692) (7, 973441680328)
  - using Lagrange polynomial
    - $l_2(x) = \frac{(x-3)(x-7)}{(2-3)(2-7)} \quad l_3(x) = \frac{(x-2)(x-7)}{(3-2)(3-7)} \quad l_7(x) = \frac{(x-2)(x-3)}{(7-2)(7-3)}$
    - $f(x) \equiv \sum_{k=\{2,3,7\}} y_k \, l_k(x) \pmod{p}$

$$\equiv 190503180520 + 482943028839\,x + 1206749628665\,x^2$$

Note: Any two shareholders cannot reconstruct the secret. For example persons 4 and 6 give their shares to each other. Any possible share from the 3rd person (say person 2) can form a distinct quadratic curve, and gives a different secret.

Function Sharing in RSA Signature

- Additive Scheme
  - RSA signature (or decryption): $s \equiv m^d \pmod{n}$
  - Additive secret splitting: $d \equiv d_1 + d_2 \pmod{\phi(n)}$
  - Alice gets d_1 and Bob gets d_2
  - Given a document m,
    - Alice signs herself and gets $s_1 \equiv m^{d_1} \pmod{n}$
    - Bob signs himself and gets $s_2 \equiv m^{d_2} \pmod{n}$
    - signature s is obtained by multiplication

$$s \equiv s_1 \cdot s_2 \equiv m^{d_1} \cdot m^{d_2} \equiv m^{d_1+d_2} \equiv m^d \pmod{n}$$

- Note:
  1. This is a non-threshold function sharing scheme. However, all the shares can be reused.
  2. In secret splitting, some trusted party must know the secret.

Function Sharing in RSA Signature

- De Santis's Scheme (using polynomial sharing)
  - RSA signature (or decryption): $s \equiv m^d \pmod{n}$
  - Shamir's polynomial secret reconstruction: $|\mathrm{I}|$ parties involved, $|\mathrm{I}| \geq \mathrm{t}$

$$p(x) \equiv \sum_{k \in \mathrm{I}} p(x_k) \prod_{\substack{i \in \mathrm{I} \\ i \neq k}} \frac{x - x_i}{x_k - x_i} \pmod{\phi(n)}$$

  - Each of the n parties gets his share x_k and $\mathrm{p}(x_\mathrm{k})$, $k = 1,2,\ldots n$
  - Given a document m, the k-th party in I does the following:
    - signs independently $s_k \equiv m^{p(x_k)\prod_{\substack{i \in I \\ i \neq k}} \frac{-x_i}{x_k - x_i}} \pmod{n}$
    - multiply together $s \equiv \prod_{k \in \mathrm{I}} s_k \equiv m^d \pmod{n}$

Function Sharing in RSA Signature

- This is a threshold function sharing scheme. All shares or the private key d can be reused for many times.
- Major **problems** of the above scheme:
  - $|I| (\geq t)$ out of n shareholders are gathered dynamically. $l_k(0)$ has to be calculated each time. This operation requires the calculation of inverse mod $\phi(n)$ and cannot be done by individual shareholder.

$$l_k(0) \equiv \prod_{\substack{i \in I \\ i \neq k}} \frac{-x_i}{x_k - x_i} \pmod{\phi(n)}$$

  - **Catastrophe: $\gcd(x_k - x_i, \phi(n)) \neq 1$**
  - One way as proposed by De Santis is to extend the group of RSA exponents to a larger set of operators. This set contains special invertible elements that do not compromise the RSA key.
  - Another way to solve this is to pre-calculate $l_k(0)$ for all possible $|I| (\geq t)$ people groups.

Correctly Sharing RSA Function

- Signature $\sigma = m^d \pmod n$
- (2, 3) sharing by a trusted dealer
  - Choose a degree-1 polynomial (a line) $f(x) = d + a\,x$
  - Share for A: $d_1 = f(1) = d + a$
    Share for B: $d_2 = f(2) = d + 2\,a$
    Share for C: $d_3 = f(3) = d + 3\,a$
    $d = 3 \cdot 2^{-1} \cdot d_1 - 2^{-1} \cdot d_3$
  - Send each person his share secretly (d, p, q are hidden from A,B,C)
- Distributed signing: A and C jointly sign the document m
  - Not working: A: $\sigma_1 \equiv m^{3\,2^{-1} d_1} \pmod n$, C: $\sigma_3 \equiv m^{-2^{-1} d_3} \pmod n$
  - working: A signs $\sigma_1 \equiv m^{3d_1} \pmod n$ and C signs $\sigma_3 \equiv m^{-d_3} \pmod n$
  - Multiply together $\sigma_1\,\sigma_3 \equiv m^{3d_1 - d_3} \equiv m^{2d} \equiv \sigma^2 \pmod n$, also we have
    $m \equiv \sigma^e \pmod n$
  - Since $\gcd(2, e)=1$, $\exists\ a, b$ s.t. $2\,a + e\,b = 1$, calculate
    $(\sigma_1\,\sigma_2)^a\, m^b \equiv \sigma^{2a}\,\sigma^{eb} \equiv \sigma^{2a+eb} \equiv \sigma \pmod n$ is the desired signature

Function Sharing in ElGamal Cryptosystem

- ElGamal cryptosystem : given p, q, $p=2q+1$, g is a generator in QR_p,

  private key: α public key: $\beta \equiv g^{\alpha} \pmod{p}$

  encryption: $k \in_R Z_p^*$, $r \equiv g^k \pmod{p}$, $c \equiv m \cdot \beta^k \pmod{p}$, $m \in QR_p$

  decryption: $m \equiv c \cdot r^{-\alpha} \pmod{p}$

- Shamir's polynomial secret splitting: $t \le |I| \le n$

$$p(x) \equiv \sum_{k \in I} p(x_k) \prod_{\substack{i \in I \\ i \neq k}} \frac{x - x_i}{x_k - x_i} \pmod{p}$$

- Each of the n parties gets his share x_k and $p(x_k)$ $k = 1,2,\ldots n$
- Given a ciphertext (r, c), the k-th party does the following:

  - decrypt independently $m_k \equiv r^{-p(x_k)\prod_{\substack{i \in I \\ i \neq k}} \frac{-x_i}{x_k - x_i}} \pmod{p}, \quad k \in I$

  - multiply together $m \equiv c \cdot \prod_{k \in I} m_k \pmod{p}$

Function Sharing in ElGamal Cryptosystem

- This is a threshold function sharing scheme. All shares of the private decryption keys can be reused for many times.
- Note:
  - t out of n shareholders are gathered dynamically. $l_k(0)$ has to be calculated each time. This operation requires the calculation of inverse mod q (since order of g and order of $r \equiv g^k \pmod{p}$ are both q) and can be done by individual shareholder.

$$l_k(0) \equiv \prod_{\substack{i \in I \\ i \neq k}} \frac{-x_i}{x_k - x_i} \pmod{q}$$

  - One thing needs to be assured before the sharing is that for all possible set I, $|I| \geq t$, $\gcd(l_k(0), q) = 1$, which is always true because q is a prime number

Blakley's Secret Sharing

- Blakley, 1979
- basic ideas: two lines in 2-dim space intersect at a 2-dim point; three planes in 3-dim space intersect at a 3-dim point; t (t-1)-dim hyperplanes in the t-dim hyperspace intersect at a t-dim point.
- (t, n) threshold sharing scheme:
  - choose a prime p, all computations will be carried out mod p
  - let s_0 be the secret to share, randomly choose t-1 random number $s_1, s_2, \ldots s_{t-1}$
  - n is the number of participants, each one gets a (t-1)-dim hyperplane passing through $(s_0, s_1, s_2, \ldots s_{t-1})$, i.e. $x_{t-1} = \sum_{i=0}^{t-2} a_i \; x_i + c$ (randomly choose a_i and choose $c = s_{t-1} - \sum_{i=0}^{t-2} a_i \; s_i$)

Blakley's Secret Sharing

- Reconstruction phase:
  - t shareholders provide their hyperplanes to deduce the secret

$$x_{t-1} = \sum_{i=0}^{t-2} a_i^{(k)} \; x_i + c^{(k)} \quad k = 1,2,...t$$

  - solving the above linear system for the secret s_0
- Note:
  - Only one coordinate should be used to carry the secret. Otherwise less than t hyperplanes are enough to solve the secret.
  - Shamir's method could be regarded as a special case
  - Shamir's method requires less information to be carried by each person: (x, y) versus t coefficients of Blakley's method.

Generalized Secret Sharing

- 8 shares are required to obtain the secret
  Boss: 2 managers:10 employees with weights of importance as 4 : 2 : 1
  - this is a special case of a (8, 18) threshold scheme
  - Boss + 2 managers can obtain the secret
- Two companies A and B share a bank vault
  - 4 employees from A and 3 from B are required to obtain the secret combination s
  - scheme:
    - write $s = s_a + s_b$
    - share s_a using a $(4, n_a)$ scheme and s_b using a $(3, n_b)$ scheme

Generalized Secret Sharing

- A, B, C, D want to share a secret, for example, using the following equation to express the reconstruction of the secret

$$F = (F_A F_B) \cup (F_B F_C) \cup (F_A F_C F_D)$$

if A and B give their shares together, F_A and F_B are both true and F is true which means that the secret can be reconstructed

- Benaloh , "Generalized Secret Sharing and Monotone Functions," Crypto'88
- Ito et. al.," Secret Sharing Scheme Realizing General Access Structure," IEEE Glob. Comm. 1987
- Harn et. al, "An 1-Span Generalized Secret Sharing Scheme," Crypto'92

Fault-Tolerant Extensions

- Sharing without trusted center
- Detecting Cheaters
- Fair reconstruction of secrets
- Verifiable secret sharing
- Robust secret sharing
- Proactive sharing

Secret Sharing Without Trusted Center

- In many applications, it is very difficult to find a trusted third party that performs the sharing.
- Solution: ex. (2, 3)-secret sharing scheme with 3 users A, B, C
  - choosing shares <u>independently</u> ex. K_A, K_B, K_C are keys of 3 locks
  - construct the main secret <u>jointly</u> ex. $K = K_A + K_B + K_C$
    i.e. put all locks on
  - every user becomes a dealer and shares his key to the other two
    $K_A \rightarrow K_{AB}$ (for user B), K_{AC} (for user C)
    $K_B \rightarrow K_{BA}$ (for user A), K_{BC} (for user C)
    $K_C \rightarrow K_{CA}$ (for user A), K_{CB} (for user B)
  - at any moment, only two users present can reconstruct the whole three keys (K_A, K_B, K_C) and therefore K

  can be generalized to recent DKG schemes

Detecting Cheaters (1/5)

- The center (the Dealer) cheats:
  - Using false threshold (use a threshold > t)
  - Sending false secret
- commitment schemes can only detect the cheating after the secret can not be recovered
- Benaloh's solution to make sure shares are *t-consistent*
  - dealer chooses a degree t-1 polynomial $h(x)$, $h(0)$ is the secret
  - dealer sends individual share to every shareholder
  - dealer chooses another 100 polynomials $g_i(x)$ and does (t, n) sharing for each polynomials
  - all n participants randomly agree 50 polynomials to recover and make sure that the degree of these polynomials is at most t-1
  - all n participants now derive together 50 $g_i(x)+h(x)$ polynomials and verify that their degrees are t-1

Detecting Cheaters (2/5)

- If degree(h(x)) = t, dealer has to choose 50 $g_i(x)$ with degree t-1 to pass the first test, and another 50 $g_i(x)$ with degree t and having the degree of $g_i(x)+h(x)$ t-1to satisfy the second test.

  The probability that he succeeds is $1 / C_{50}^{100}$

- Sending false secret can be prevented by VSS Schemes

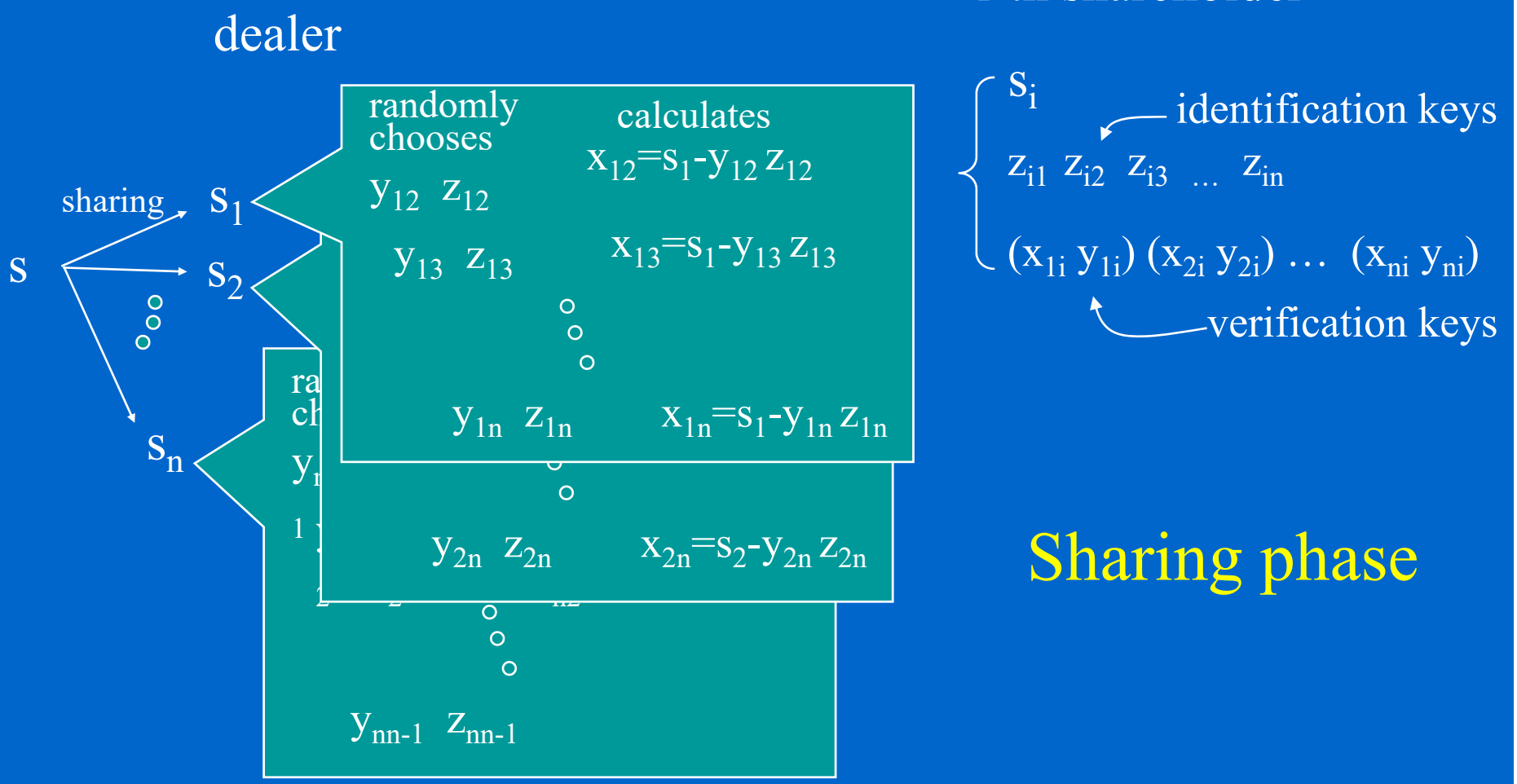
Detecting Cheaters (3/5)

A player sends false share to prevent reconstruction of the secret.

- Tompa and Woll, "How to share a secret with cheaters," J. of Cryptology 1988
  - for Shamir's method
- Rabin, "Robust Sharing of Secrets When the Dealer is Honest of Cheating," MS Thesis, Hebrew Univ.
  - the i-th shareholder receives from the dealer
    - a share s_i (also satisfy $s_i = x_{ij} + y_{ij}\, z_{ij}$ for all j=1,2,…,n)
    - n-1 identification keys z_{ij} for proving the correctness of his share to others
    - n-1 verification key pairs (x_{ji}, y_{ji}) for verifying other's shares
- Ben-Or et. al. "Completeness Theorems for Non-cryptographic Fault-tolerant Distributed Computation," ACM STOC'88
  - Using Error Correcting Code: (t, n) scheme: the secret can be reconstructed from the n shares up to t false or missing shares if $n \geq 3t+1$.

Detecting Cheaters (4/5)

✧ Rabin's scheme illustrated:



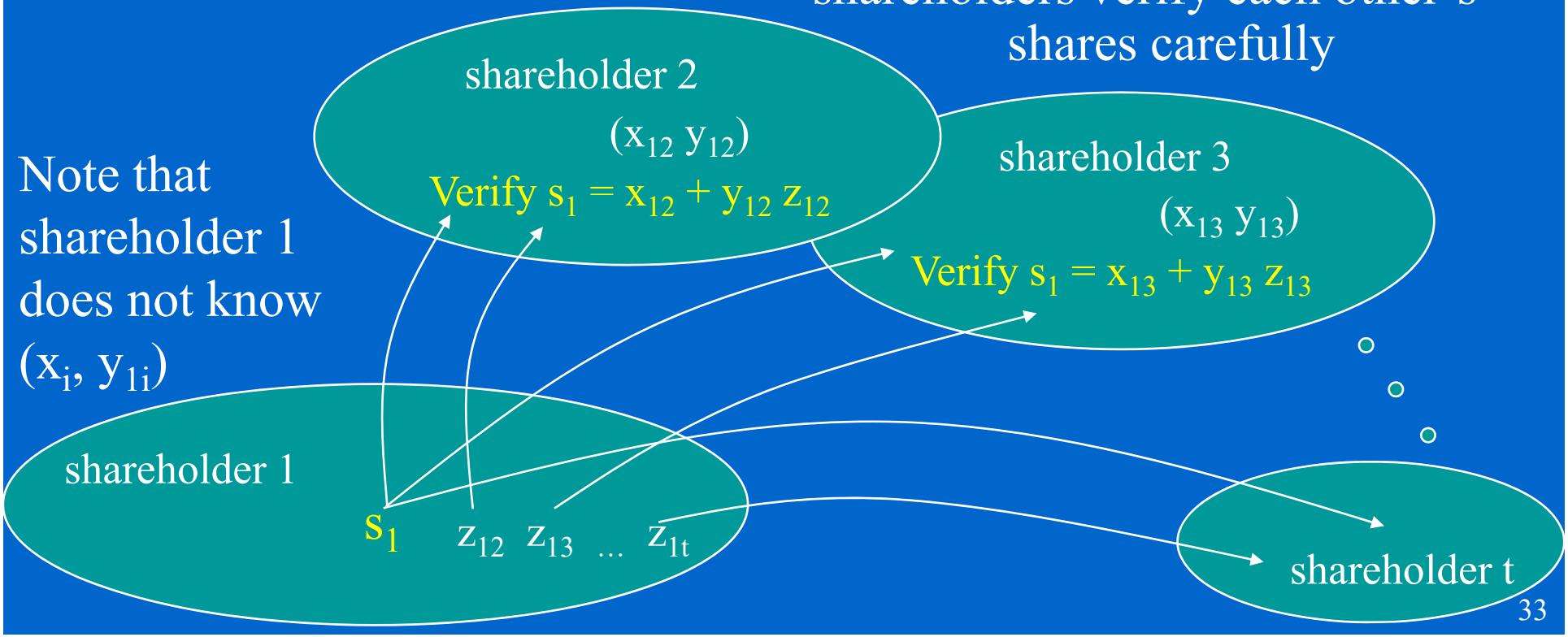
Sharing phase

Detecting Cheaters (5/5)

- Rabin's scheme illustrated:

Reconstruction phase

t shareholders announce their shares sequentially,
shareholders verify each other's shares carefully



Fair Reconstruction of a Secret

- Even we have Rabin's mechanism to detect the cheating shareholder, the last one providing the share can always reconstruct the secret before he sends his share to others.
- Solution: (Lin and Harn, "Fair reconstruction of a secret," IPL 1995 vol 55)
  - dealer hides the key K and a sentinel S in a sequence

    $D^{(1)}, D^{(2)}, \ldots, D^{(j-1)}, K, S, D^{(j+2)}, \ldots D^{(m)}$

    S is known to everyone, $D^{(i)}$ are random values
  - dealer shares each number independently to all n shareholders
  - at the reconstruction phase:
    - Everyone has to follow the protocol correctly. If anyone cheats, the protocol aborts.
    - Before reconstructing S, no one knows that the previous reconstructed one was the real secret K.

Robust Secret Sharing

- Adversarial shareholders: *prevent good shareholders from reconstructing the secret*
- In Shamir's (t, n) sharing scheme: one adversary in the t shareholders may lie about the value of his share.
  - Prevent the reconstruction of the secret
  - other shareholders will not know the secret is fake
  - other shareholders will not know who is to blame
- Solution: Shareholders prove that their computations and/ or communications follow protocols correctly while keep their shares confidential.
  - Zero knowledge proof: any protocol statement can be expressed in a language in NP; each language in NP has a ZKP

Verifiable Secret Sharing

- Dealer provides shares privately to each parties.
  - No individual knows whether the share he gets is correct or not unless *t* users reconstruct the secret.
  - Till then, the shares are assured to be correct but also useless.
- How to assure each shareholders
  - the shares they obtained can be used to reconstruct the correct secret?
  - at least construct one unambiguous secret?
- One method is by cross verifying shares in several secret sharing schemes (primary and secondary shares)
  - Chor 85, Feldman 87, Pedersen 91

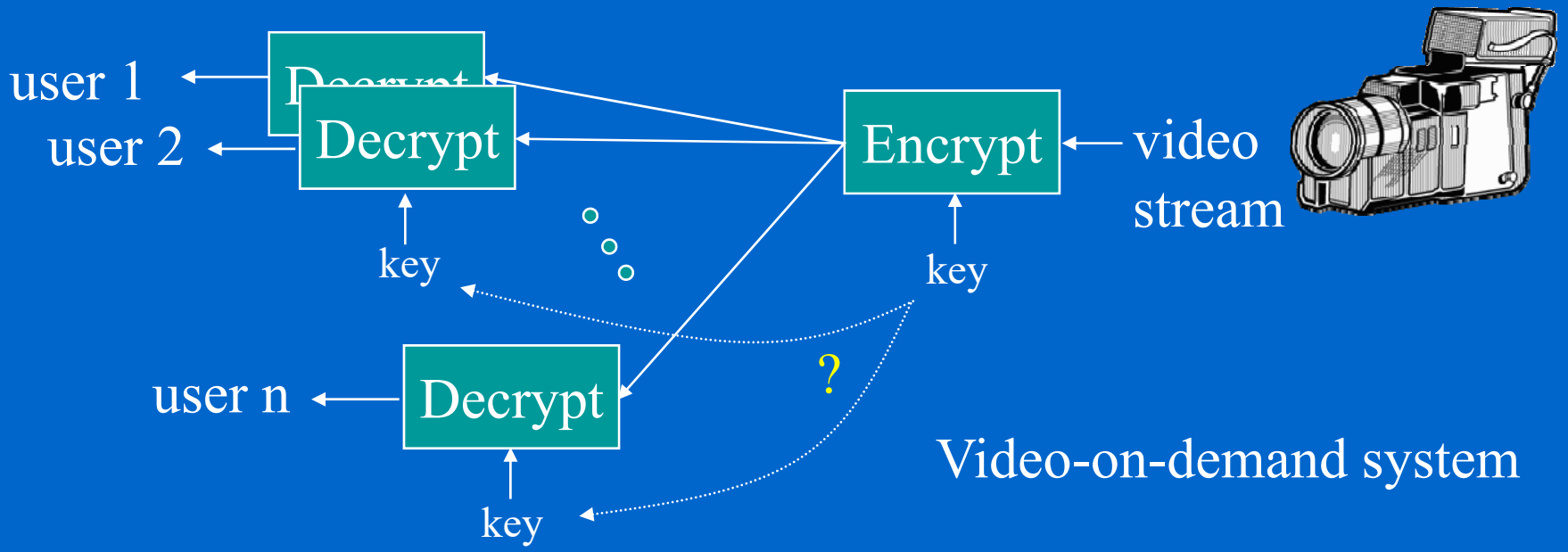
Public Verifiable Secret Sharing

- In a VSS scheme, the participants can verify the validity of only their own shares, but they cannot know whether other participants have also received valid shares.
  - Note: in Benoloh's verification for t-consistency of all shares, all participants are assured that their shares can reconstruct a unique secret. It is a PVSS scheme.
- In a PVSS scheme, a public-key encryption function $E_{k_i}(\cdot)$ is used not only to distribute the shares to each participants but also to publish these encrypted shares for providing a ZKP that every shareholder has the correct share.

Proactive Sharing

- Mobile adversary: may occupy only up to k-1 shareholders at any time, but it may occupy any or all shareholders over the lifetime of the system.
  - Ex.: computer virus, hackers, disgruntled ex-employees
- Proactive threshold secret sharing protocols protect against mobile adversaries.
  - Proactive signature schemes such as DSS, Schnorr, ElGamal and RSA

Secret Broadcasting (1/5)



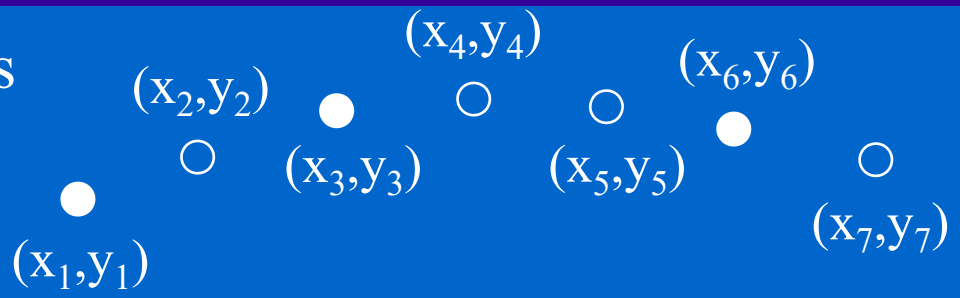
- Requirements:
  - the amount of video data is huge, encryption/decryption should be fast
  - users are dynamically grouped according to their subscriptions to a particular program
  - the exchange of keys should be fairly quick

Secret Broadcasting (2/5)

- Basic scheme:
  - broadcaster chooses a random key S
  - broadcaster distributes securely the key S to all subscribers to the program using a secret key system or a public key system
  - broadcaster encrypts the program with E_S(video) and broadcast
  - only subscribers can decrypt the program D_S(video)
- Problems:
  - broadcaster must do all the encryption for distributing S before the program can start (what if the number of subscribers are 10000?)
  - broadcaster must have a second channel to communicate with every intended recipient (instead of the broadcast channel)

Secret Broadcasting (3/5)

3 recipients, 4 non-recipients with fixed secrets – pseudoshares



k=3 { (x_1,y_1), (x_3,y_3), (x_6,y_6)

choose j=5 randomly choose { (x_8,y_8), (x_9,y_9), (x_{10},y_{10}), (x_{11},y_{11}), (x_{12},y_{12})

introduce a degree of randomness in case of resending S

$(0, S)$

Calculate a degree k+j=8 polynomial using Lagrange interpolation polynomial

Broadcast a new set of randomly chosen k+j=8 shares

(x_{13},y_{13}) (x_{14},y_{14}) (x_{15},y_{15}) (x_{16},y_{16}) (x_{17},y_{17}) (x_{18},y_{18}) (x_{19},y_{19}) (x_{20},y_{20})

Recipient (ex. user 1) receives (x_{13},y_{13}) (x_{14},y_{14}) (x_{15},y_{15}) (x_{16},y_{16}) (x_{17},y_{17}) (x_{18},y_{18}) (x_{19},y_{19}) (x_{20},y_{20}) together with his own private share (x_1,y_1) can reconstruct the key S while non-Recipient do not have enough information to reconstruct.

Secret Broadcasting (4/5)

- Broadcast using Shamir's secret sharing scheme

  To broadcast to k recipients

  1. Choose $j \geq 0$
  2. Create a k+j+1 out of 2k+2j+1 secret sharing system
     a. secret = S
     b. pseudoshares of recipients as real shares
     c. pseudoshares of non-recipients must not be real shares
     d. broadcaster includes j randomly chosen, unassigned pseudoshares
  3. Broadcast k+j randomly chosen shares - all different from step 2
  4. Each subscriber adds his pseudoshare as a possible share to the k+j shares received
     a. if that pseudoshare is a real share, as in step 2b, he recovers S
     b. if not, as in step 2c, he does not recover S

Secret Broadcasting (5/5)

- Problem I: after one round of broadcast, each recipient can deduce the secrets of other recipients
  - Solution: x_i should also be secret
- Problem II: After two broadcasts, two different curves are chosen to share two secret keys to two sets of subscribers. If the intersection of these two sets are not empty, these two curves must intersects at their secrets. Therefore, whoever subscribes both broadcasts can solve other common members' secrets.
  - Solution: ??
- Problem III: Why using those extra j points? Only for resending the same secret.

References

- General
  - A. Shamir, “How to share a secret,” Comm. ACM 1979, pp. 612-613
  - R. Blakley, “Safeguarding cryptographic keys,” FIPS Conf Proc. 1979, pp. 313-317
  - P. S. Gemmell, “An Introduction to Threshold Cryptography,” CryptoBytes 97
  - Y. Desmedt and Y. Frankel, “Threshold Cryptosystems,” Crypto’89
  - Y. Desmedt, “Threshold Cryptography,” European Transactions on Telecomm. 1994, pp. 449-457
  - Y. Desmedt and Y. Frankel, “Shared generation of authenticators and signatures,” Crypto’91
  - T. P. Pedersen, “A threshold cryptosystem without a trusted party,” Eurocrypt’91
  - S. R. Blackburn, M. Burmester, Y. Desmedt, and P. R. Wild, “Efficient Multiplicative Sharing Schemes,” Eurocrypt’96

References

- Verifiable Secret Sharing
  - B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch,"Verifiable Secret Sharing and Achieving Simultaneous Broadcast," FOCS 1985, pp. 335-344
  - P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," FOCS'87
  - T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," STOC 1989
  - R. Gennaro, S. Jarecki, H. Krwczyk, and T. Rabin, "Robust threshold DSS signatures," Eurocrypt'96
  - T. P. Pederson, "Non-interactive and information-theoretic secure verifiable secret sharing," Crypto'91
- Multi-secret Sharing
  - W. Jackson, K. Martin, and C. Okeefe, "Multi-secret Threshold Schemes," Crypto'93

References

- Function Sharing
  - A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," STOC'94
  - Y. Frankel, P. Gemmell, and M. Yung, "Witness-based Cryptographic Program Checking and Robust Function Sharing," STOC'96
- Robust Secret Sharing, Distributed Key Generation
  - T. Rabin, "Robust sharing of secrets when the dealer is honest or faulty," JACM 41
  - R. Gennaro, S. Jarecki, H. Krwczyk, and T. Rabin, "Robust threshold DSS signatures," Eurocrypt'96
  - R. Genero, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and Efficient Sharing of RSA Functions," Crypto'96

References

- Proactive Secret Sharing
  - Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung. “Proactive RSA,” http://www.cs.nsandia.gov/psgemme/crypto/rpro.html
  - A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung, “Proactive Public Key and Signature Systems,” http://theory.lcs.mit.edu/cis/cis-publications.html
  - A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, “Proactive secret sharing, or: how to cope with perpetual leakage,” Crypto’95
  - S. Jarecki, “Proactive Secret Sharing and Public Key Cryptosystems,” Master Thesis, MIT 1996.
  - N. Alon, Z. Galil, and M. Yung, “Dynamic re-sharing verifiable secret sharing against a mobile adversary,” European Sym. On Algorithms 1995 , LNCS 979
- Broadcast
  - S. Berkovits, “How to broadcast a secret,” Eurocrypt 91
  - A. Fiat and M. Naor, “Broadcast Encryption,” Crypto 93
  - J. Horwitz, “A Survey of Broadcast Encryption,” 2003