Secret Sharing

CS395T Design and Implementation of Trusted Services

Ankur Gupta

Overview of the Talk

- Hugo Krawczyk. Secret Sharing Made Short, 1993.
- Josh Cohen Benaloh. Secret Sharing Homomorphisms: Keeping Shares of A Secret Secret, 1987.
- Josh Cohen Benaloh and Jerry Leichter. Generalized Secret Sharing and Monotone Functions, 1990.

Secret Sharing Made Short

Hugo Krawczyk 1993

- Motivation: Shamir's (n, m) threshold scheme requires that each secret share be as long as the secret to be shared.
- Therefore, for a secret S of length r, total length of share is rn.
- We present an optimal scheme that requires each secret share to be only r/m + a constant (independent of r, n) in size.

Perfect Secrecy

- Perfect Secrecy

  1) In Shamir's (n, m) threshold scheme, no information about the secret is revealed in the information theoretic sense. If S_0 is the secret, for every $k < m$, let $S_1, \ldots, S_k$ be any k shares then

  $$Pr(S_0 \mid S_1, \ldots, S_k) = Pr(S_0)$$

  2) No bounds on the computation power of adversary are assumed.

Computational Secrecy

- Computational Secrecy:
  a) Adversary is resource bounded.
  b) No information is revealed using only polynomial resources.

Polynomial Indistinguishability: Two probability distributions are polynomial time indistinguishable if any probabilistic polynomial time algorithm behaves essentially the same when its input is selected from either of the two distributions.

Computation Secrecy: Formal definition

- An (n, m) threshold scheme is computationally secure if for any two secrets S′ and S″, for any k < m, the distributions on shares $D(S'; S'_1, \ldots, S'_k)$ and $D(S''; S''_1, \ldots, S''_k)$ introduced by the scheme are polynomially indistinguishable.

Ingredients of the Scheme

- (n, m) Information Dispersal Scheme (IDS) introduced by Rabin. A piece of information (say a file F) is divided into n shares and transmitted over an unreliable channel. Any m shares suffice to reconstruct the file F. Size of each share is |F|/m.
- Secure private key encryption (ENC)
- Perfect (n, m) Secret Sharing Scheme (PSS) like Shamir's threshold scheme.

Distribution Scheme

1) Choose a random encryption key K. Let $E=ENC_K(S)$ where S=secret
2) Using IDS partition E into n fragments $E_1, E_2, \ldots, E_n$.
3) Using PSS generate n shares of the key K: $K_1, K_2, \ldots, K_n$.
4) Send to participant P_i, the share (E_i, K_i) privately.

Reconstruction Scheme

1) Collect from any m participants P_j, $1<=j<=m$, their shares $S_j=(E_j, K_j)$.
2) Using IDS, reconstruct E from E_js.
3) Using PSS reconstruct K from K_js.
4) Decrypt E using K to recover the secret S.

Analysis

- Length of each share $S_i=(|S|/m, |K|)$.
- Correctness (Sketch):

a) m-1 E_js reveal no more information about S than E itself (by security of ENC).

b) m-1 shares reveal absolutely no information about the encryption key K (by security of PSS).

Robust Secret Sharing

- How to ensure recovery of secret in presence of malicious participants?
- Solution: Let the dealer digitally sign all the shares.
- Now a participant can't cheat!

Secret Sharing Homomorphisms: Josh Benaloh 1987

- Motivation:

1) Alice distributes a secret A to n agents using an (n, m) scheme.
2) Bob distributes secret B to the same n agents using an (n, m) scheme.
3) How can any m of the n agents determine A + B while revealing as little about A and B as possible.

Homomorphism Property

- Let F: T^m -> S be a (n, m) threshold scheme. If $D_{i1}, D_{i2}, \ldots, D_{im}$ be any m shares then the secret D is

  $D=F(D_{i1}, D_{i2}, \ldots, D_{im})$
- (n, m) is (+, *) homomorphic, if

  $D=F(D_{i1}, D_{i2}, \ldots, D_{im})$

  $D'=F(D'_{i1}, D'_{i2}, \ldots, D'_{im})$ then

  $D+D'=F(D_{i1}*D'_{i1}, \ldots, D_{im} * D'_{im})$

(+, *) Composite Scheme

- Is an (n, m) scheme which divides a set of s secrets $d_1, \ldots, d_s$ into subshares $d_{i,j}$ $1<= i <= n$, $1<= j <= s$, such that

1) The super-secret $D=d_1+d_2+\ldots+d_s$ is easily computable from the super-shares $D_i=d_{i,1}*\ldots*d_{i,s}$
2) $Pr(d_j=x_j \mid D=X) = Pr(d_j=x_j \mid D=X, 1<=i<=n, D_i=X_i$, for all i' in I', $1<=j<=s$, $d_{i',j}=x_{i',j})$ for every I', |I'| <m i.e. knowing all super-shares, super-secret and at most m-1 sub-shares of each sub-secret reveals no information about every sub-share

Homomorphism and Compositeness

- Theorem: If |S|=|T| are finite then every (+, *) homomorphic threshold scheme is a (+, *) composite threshold scheme.
- Proof (sketch): Assume s=2 I.e. there are only two sub secrets A and B. Let S=A+B. Further let k=m-1 conspire together so that $a_1,\ldots, a_k$ and $b_1,\ldots, b_k$ are known. Therefore $s_1,\ldots, s_k$ (where $s_i=a_i+b_i$) are also known. As S is known, and since |S|=|T| (and finite), therefore F^{m-1}:T->S is 1-1 and hence $s_{k+1}, \ldots, s_n$ can be computed. Thus knowing them beforehand gives no additional information.

Examples

- Shamir's scheme is (+, +) homomorphic but not (x, x) or (x, +).
- A variation of Shamir scheme is (x, +) homomorphic (secret=g^a, a is divided into n subshares using the Shamir scheme).
- By above theorem they are also composite schemes. Hence composition of sub-secrets can be obtained without revealing the sub-secrets.

Applications

- Verifiable Secret Sharing
  - Will use (+,x) homomorphism property of an encryption scheme and (+,+) composite scheme.
- Secret Ballot Election
  - Will use a (+,+) composite scheme.

Encryption Scheme

- Verifiable secret sharing scheme will use the following encryption scheme:
- Let r > |S| be prime, p and q also prime s.t. r divides p-1 but not q-1. N=pq. Let y be such that gcd(N, y) =1 and y is NOT the r^{th} residue mod N (I.e. $y \neq a^r$ mod N, for any a). (N, y) are made public. To encrypt a secret s, choose a random x (relatively prime to N) and let $E(s,x,y,N)=y^s x^r$ mod N.
- Knowing p and q, s can be easily determined.
- E is (+, x) homomorphic.

Verifiable Secret Sharing

- Objective: Dealer should be able to prove the participants that he made a consistent deal I.e. from every k sub-shares we get the same secret s, without revealing the secret s.
- This is useful in voting schemes where a voter needs to prove that he cast a valid vote.
- We assume Shamir's scheme as the secret sharing scheme (and hence a (+,+) composite scheme).

Solution: Interactive Proof

- Interactive Proof (Probabilistic)
- Dealer will convince the participants that he made a consistent deal with high probability.
- For Shamir's scheme, this boils down for dealer to convince the participants that he used at most d=m-1 degree polynomial P.

Algorithm

1. Encryptions of the values of the points that describe P are released by dealer.
2. Similar encryptions of 100 more random polynomials of degree at most d are released to the verifiers.
3. A random subset of the random polynomials is selected by the verifiers.
4. The chosen subset of polynomials are decrypted by the prover and shown to verifiers. All these polynomials be of at most degree d.
5. Each remaining polynomial is added to P. Each of these sum polynomials is decrypted by prover. They all must be of degree at most d.

Proof of Correctness

- Fact: If sum of two polynomials is of degree at most d, then either both polynomials are of degree at most d or both are of degree greater than d.
- Revealing a random subset of the set of random polynomials gives the confidence that the remaining polynomials are each of degree at most d.
- Since sum of each of the remaining polynomials with the polynomial P is of degree at most d, therefore P itself is of degree at most d.
- The homomorphism of E and Shamir's scheme helps in guaranteeing that sum of secrets can be revealed without revealing the constituent secret polynomial.

Secret-Ballot Voting

- Each voter votes 0 or 1 (yes or no).
- N independent organizations hold the election. Assumption: at most m-1 of them collude.
- Voter uses an (+,+)-composite (n, m) threshold scheme and sends the shares to the organizations.
- After election, each organization sums up the shares that it received from different voters.
- Any m organization can get the vote count without compromising secrecy of each voter's vote.

Generalized Secret Sharing and Monotone Functions:

Josh Benaloh and Jerry Leicheter 1990

- Motivation: Let a secret S be shared among P, a set of trustees, such that any qualified subset of trustees is able to recover the secret and no unqualified subset of trustees is able to get any information about the secret.
- Example: Let access structure be $Q=\{\{a\},\{b,c\},\{d,e,f,g\}_3\}$ I.e. either a can recover secret, or b and c together or any 3 of {d,e,f,g} together can recover secret.

Monotonic Access Structures

- The access structure must be monotonic.
- Monotonic Access Structure: Let ∏ be any family of set of subsets of P. ∏ is said to be monotonic if for all non-threshold sets A:

$$A \in \Pi, A \subseteq A' \Rightarrow A' \in \Pi$$

Facts

- Every monotone access structure can be represented by a boolean formula (containing only "and" and "or" and threshold gates) where each variable v_i in formula corresponds to a participant P_i.
- Therefore, it suffices to show how the secret should be shared across "and", "or" and threshold gates.
- Example: The access structure Q given earlier can be written as:

$$Q = a \vee (b \wedge c) \vee Thres3(d,e,f,g)$$

Impossibility Result

- Theorem: There exists monotone structures for which no (n, m) threshold scheme exists.
- Similar to Quorum Systems vs Voting assignments
- Proof: Consider the access structure ∏ defined by

$$((A \wedge B) \vee (C \wedge D))$$

- Let a, b, c and d denote the weights of each participant and the threshold scheme be (n, t) where n=a+b+c+d. Then a+b>=t, c+d>=t. WLOG, let a>=b and c>=d. Then a>=t/2, c>=t/2, therefore a+c>=t. Hence A and C together can determine secret S. Not allowed! Hence no such (n, t) threshold scheme exists.

Generalized Secret Sharing

- Definition: Given a set P and a monotone access structure ∏, a generalized secret sharing scheme divides a secret s into shares $s_{i,j}$ such that :

1. When A is in ∏, s can be reconstructed from the shares $s_{i,j}$ in A.
2. For A not in ∏, shares $s_{i,j}$ in A give no information about secret s.

- Notation: Let $T_m(s;p_1,..,p_n)$ denote a (n, m) threshold scheme.

Intuition

- Suppose a secret s, $0 <= s < r$, needs to be shared between P_1 and P_2 s.t.:

a) Either of them should be able to determine the secret. Scheme: give s to both of them

b) Only both of them together should be able to determine s: choose s_1 and s_2 randomly s.t. $s = s_1+s_2 \mod r$. Give s_1 to P_1 and s_2 to P_2.

Scheme

- Let T(s,F) be our generalized secret scheme. We define it recursively:

1. $T(s,v_p)$=assign share s to p.
2. $T(s, Or(F_1,\ldots,F_n))=T(s, F_1),\ldots,T(s, F_n)$ I.e divide s as s to all formula F_i
3. $T(s, And(F_1,\ldots,F_n))=T(s_i, F_i)$ $1<=i<=n$ where $s=\sum_{1<=i<=n} s_i$, s_i being random
4. $T(s,THRES_m(F_1,\ldots,F_n))= T(s_i, F_i)$ $1<=i<=n$ where $s=T_m(s;f_1,..,f_n)=[s_i]_{1<=i<=n}$

Proof of Correctness

- We use structural induction on the length of access structure formula.
- Base case is trivial.
- Let the monotone formula f be of form $Or(F_1,\ldots,F_n)$. Since s is divided over each F_i independently (by using $T(s, F_i)$) therefore for any A not in ∏, no joint information is conveyed by shares that belong to different $T(s, F_i)$.
- $F=And(F_1,\ldots,F_n)$. If A not in ∏ then there is an i s.t. shares of $T(s_i, F_i)$ in A give no information about s_i. Since all s_j were chosen randomly s.t. $s=\sum s_j$ therefore if s_i is not known then no information about s is revealed.
- Similar argument applies for the threshold operator.

Homomorphism

- Theorem: If the $T_m(s;p_1,..,p_n)$ is (+,+) homorphic then the generalized secret sharing scheme is also (+,+) homomorphic.

Limitations

- Theorem: There exists monotone access structures for which the scheme is not efficient (formula size and hence number of shares is not polynomial in n)
- Proof (Sketch): Combinatorial Argument: There are doubly exponential monotonic access structures whereas there are only exponentially many polynomial sized access structures corresponding to polynomial sized formulae.