Function Sharing Schemes based on the Chinese Remainder Theorem

Kamer Kaya, Ali Aydın Selçuk, Zahir Tezcan

Department of Computer Engineering
Bilkent University

Outline

- Threshold cryptography
- Secret sharing schemes
- Function sharing problem
- A function sharing scheme based on the CRT
- Conclusions

Overview

Threshold cryptography:

- Secret sharing: Three major schemes
  - Shamir
  - Blakley
  - Asmuth-Bloom

- Function sharing
  - Traditionally based on Shamir; occasionally on Blakley.
  - So far no solutions with Asmuth-Bloom.

This work: Achieving function sharing using the A-B SS scheme.

Threshold Cryptography

Secret Sharing Problem:
How to share a sensitive secret d among n parties s.t. only a certain number t of them can together construct the secret?

Applications:
– Storage of sensitive cryptographic keys (e.g. root key in a PKI system)
– Command & control of nuclear weapons

E.g. An (n, n) secret sharing scheme:
To share an ℓ-bit secret key d,
– generate random ℓ-bit y_i, $i = 1, \ldots, n-1$,
– $y_n = d \oplus y_1 \oplus y_2 \oplus \ldots \oplus y_{n-1}$
– give y_i to the ith user.

This scheme is *perfect*: A coalition smaller than t obtains no information.

Q: How to generalize to arbitrary (t, n)?

Shamir Secret Sharing Scheme

Shamir's (t, n)-threshold scheme:
– Choose large prime $p > d, n$
– Generate random polynomial in $\mathbb{Z}_p[x]$

$$f(x) = a_{t-1}x^{t-1} + \ldots + a_1 x + a_0$$

where $a_0 = d$.
– Share of user i is $y_i = f(x_i)$ for a random x_i.

Secret construction by Lagrange interpolation:

$$f(x) = \sum_{i \in \mathcal{S}} y_i \prod_{\substack{j \in \mathcal{S} \\ j \neq i}} \frac{x - x_j}{x_i - x_j}$$

for $|\mathcal{S}| \geq t$.

Remark: Shamir SSS is perfect.

Blakley Secret Sharing Scheme

Fact: In t-dimensional space, a system of t (non-degenarate) hyperplanes intersect at a single point.

Blakley's (t, n)-threshold scheme:
– Choose large prime $p > d, n$
– Choose a random point $x' = (x'_1, \ldots, x'_t)$ in $\mathbb{Z}_p^t$ s.t. $x'_1 = d$.
– Give the ith user a random hyperplane

$$a_{i1}x_1 + \ldots + a_{it}x_t = y_i,$$

$1 \leq i \leq n$, each passing through x'.

Secret construction is by solving the system

$$A_{\mathcal{S}}\, x = y_{\mathcal{S}}.$$

Remark: Shamir's is a special case where

$$a_{ij} = \alpha_i^{j-1}$$

for some random α_i.
I.e., where each $A_{\mathcal{S}}$ is Vandermonde.

Function Sharing Problem

Combiner platform may not be trusted.
Users would rather not reveal their shares.

Can we share the function rather than the private key, so that each user can compute his part without revealing his share?

E.g.: An (n, n) RSA signature sharing:

$$d = d_1 + \ldots + d_n \bmod \phi(N)$$

where d_i, $i \leq n - 1$, are randomly generated.

To compute signature $w^d \bmod N$, the ith user computes $w^{d_i} \bmod N$.

The partial results are combined as

$$\begin{aligned} \prod_i w^{d_i} \bmod N &= w^{\sum_i d_i} \bmod N \\ &= w^d \bmod N \end{aligned}$$

Q: How to generalize to arbitrary (t, n)?

Function Sharing with Shamir

Note that

$$d = f(0) = \sum_{i \in \mathcal{S}} y_i \prod_{\substack{j \in \mathcal{S} \\ j \neq i}} \frac{0 - x_j}{x_i - x_j}$$

Hence,

$$w^d = \prod_{i \in \mathcal{S}} w^{y_i \prod_j \frac{0 - x_j}{x_i - x_j}}$$

where x_i are publicly known among members; y_i are secret.

Only if we could do Lagr.int. in $\mathbb{Z}_{\phi(N)}$:

- $(x_i - x_j)^{-1}$ may not exist in $\mathbb{Z}_{\phi(N)}$.

- Moreover, $\phi(N)$ should not be known.

Solutions to FS with Shamir

Ingenious solutions needed:

- *Desmedt & Frankel (1991)*
  – Dealer computes all inverses.
  – Distributes them mixed with shares.
  – Not so efficient.

- *De Santis, Desmedt, Frankel, Yung (1994)*
  – Arithmetic is taken to $R = \mathbb{Z}(u)$ where u is a cyclotomic root of prime order.
  – $x_i \in R$ are chosen s.t. inverses exist for all $(x_i - x_j)$.
  – Inverses computed in $\mathbb{Z}(u)$ are also inverses in $\mathbb{Z}_{\phi(N)}(u)$.

- *Shoup (2000)*
  – RSA function is slightly modified.
  – No need to take inverses nor know $\phi(N)$.
  – Very simple and practical.

Asmuth-Bloom Secret Sharing Scheme

Secret Sharing:

1. Choose integers $m_0 < m_1 < \ldots < m_n$ s.t.
   - m_i are relatively prime
   - $m_0 > d$ is a prime
   - m_i satisfy (for perfectness)

$$\prod_{i=1}^{t} m_i \; > \; m_0 \prod_{i=1}^{t-1} m_{n-i+1}$$

2. Let $M = \prod_{i=1}^{t} m_i$. Compute

$$y = d + am_0$$

   where a is some random integer s.t. $0 \leq y < M$.

3. Share of the i^{th} user is

$$y_i = y \bmod m_i.$$

Asmuth-Bloom SSS

Secret Construction:

1. Given the system

$$y \equiv y_i \pmod{m_i}$$

for $i \in \mathcal{S}$, solve y in $\mathbb{Z}_{M_{\mathcal{S}}}$ using the CRT.

(Note that $M_{\mathcal{S}} \geq M$ for all valid $\mathcal{S}$.)

2. Compute the secret as

$$d = y \bmod m_0.$$

Function Sharing with Asmuth-Bloom

First, an additive solution for CRT:

1. Let $M'_{\mathcal{S},i} = M^{-1}_{\mathcal{S}\setminus\{i\}} \pmod{m_i}$. User i computes

$$u_i = y_i M'_{\mathcal{S},i} M_{\mathcal{S}\setminus\{i\}} \bmod M_{\mathcal{S}}.$$

2. y is computed as

$$y = \sum_{i\in\mathcal{S}} u_i \bmod M_{\mathcal{S}}.$$

3. The secret d is computed as

$$d = y \bmod m_0.$$

Note that m_0 is not needed until Step 3.

Also note that m_0 need not be prime.

Sharing RSA with Asmuth-Bloom

Secret construction is additive

$$\sum_{i \in \mathcal{S}} y_i M'_{\mathcal{S},i} M_{\mathcal{S}\setminus\{i\}} \bmod M_{\mathcal{S}}$$

hence may be suitable to share RSA:

$$w^d \bmod N = \prod_{i \in \mathcal{S}} w^{y_i \cdots} \bmod N$$

Challenge: But how to include (mod $M_{\mathcal{S}}$) in the exponent?

(Shamir and Blakley works with the same modulus p for all coalitions, which was made related to $\phi(N)$. Here, each coalition has a different $M_{\mathcal{S}}$. How to relate $M_{\mathcal{S}}$ to $\phi(N)$?)

A FSS for RSA Signatures

1. RSA setup with $p = 2p' + 1$, $q = 2q' + 1$.
   $N = pq$; $ed \equiv 1 \pmod{\phi(N)}$.
   Use A-B to share d with $m_0 = \phi(N) = 4p'q'$.

2. To sign w, user $i \in \mathcal{S}$ computes

$$u_i = y_i M'_{\mathcal{S},i} M_{\mathcal{S}\setminus\{i\}} \bmod M_{\mathcal{S}},$$
$$s_i = w^{u_i} \bmod N.$$

3. The *incomplete signature* $\overline{s}$ is

$$\overline{s} = \prod_{i \in \mathcal{S}} s_i \bmod N.$$

4. Let $\lambda = w^{-M_s} \bmod N$ be the *corrector*. Try

$$(\overline{s}\lambda^j)^e = \overline{s}^e (\lambda^e)^j \stackrel{?}{\equiv} w \pmod{N} \qquad (1)$$

   for $0 \leq j < t$. Then the signature s is

$$s = \overline{s}\lambda^\delta \bmod N$$

   where δ is the j value satisfying (1).

A FSS for RSA Signatures

Remarks:

1. In the exponent,

$$\sum_{i \in \mathcal{S}} u_i = y + \delta M_{\mathcal{S}}$$

for some $1 \leq \delta \leq t-1$.
We add in the corrector $\lambda = w^{-M_{\mathcal{S}}}$ until we obtain w^y.

2. When we obtain y in the exponent, we have

$$w^y \equiv w^d \pmod{N}$$

since $y = d + am_0$ where $m_0 = \phi(N)$.

3. Note that $x^e \bmod N$ is a bijection in $\mathbb{Z}_N$. Hence there is a unique $s = \overline{s}\lambda^j$ that satisfies $s^e \bmod N = w$.

Sharing ElGamal Decryption and Signature

– Large prime p; generator $g \in \mathbb{Z}_p^*$
– Private key $\alpha \in \{0, \ldots, p-1\}$ (shared)
– Public key $\beta = g^\alpha \bmod p$

Decryption:
– Somewhat similar to RSA.
– We need to compute $x^\alpha \bmod p$.

Signature:
To sign message w, we need to compute

$$r = g^k \bmod p$$
$$s = (w - r\alpha)k^{-1} \bmod (p-1)$$

where k is a random integer jointly generated, and has to remain secret throughout.

Comparison to Earlier Works

- *Performance:*
  – User: $t+1$ multiplications, one inversion, one exponentiation
  – Combining: $t-1$ multiplications
  – Correction: one exponentiation, up to $t-1$ multiplications

- Better than Desmedt & Frankel (1991) and De Santis et al. (1994).

Concluding Remarks

Up until now, FS has traditionally been based on Shamir or Blakley. We showed how it can be done with Asmuth-Bloom.

The schemes are comparable to the earlier works performance-wise.

The proposed schemes may be more suitable at places as it does not require computing inverses mod $\phi(N)$, which may be problematic or sometimes impossible.