Robust Threshold Schemes Based on the Chinese Remainder Theorem

Kamer Kaya and Ali Aydın Selçuk

Department of Computer Engineering
Bilkent University

Outline

- Threshold cryptography
- Secret sharing problem
- Function sharing problem
- CRT based function sharing schemes
- Robustness in CRT based function sharing
- Conclusions

Threshold Cryptography

- Secret sharing: Two major schemes
  - Shamir
  - Blakley

  and the rest, e.g., Asmuth-Bloom.

- Function sharing
  - Traditionally based on Shamir.
  - Occasionally on Blakley.
  - Lately, solutions with CRT.

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 of size smaller than t obtains no information.

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

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 and reveal his partial result 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)?

CRT Based Function Sharing

Asmuth-Bloom 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)

$$\prod_{i=1}^{t} m_i \; > \; {m_0}^2 \prod_{i=1}^{t-1} m_{n-i+1}$$

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

$$y_i = y \bmod m_i.$$

CRT Based Function Sharing

Sharing RSA with Asmuth-Bloom

Let $\mathcal{S}$ be a coalition of size t. Let $M_{\mathcal{S}} = \prod_{i\in\mathcal{S}} m_i$ and $M'_{\mathcal{S},i} = M^{-1}_{\mathcal{S}\setminus\{i\}} \pmod{m_i}$.

Secret construction is additive:

$$u_i = y_i M'_{\mathcal{S},i} M_{\mathcal{S}\setminus\{i\}}$$
$$y = \left(\sum_{i\in\mathcal{S}} u_i\right) \bmod M_{\mathcal{S}}$$
$$d = y \bmod m_0$$

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?

CRT Based Function Sharing

The Correction Procedure

- In the RSA signature setting with the public private key pair (e, d), the ith user contributes

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

- When the combiner multiply the partial results he obtains an incomplete signature

$$\begin{aligned} \overline{s} &= \prod_{i \in \mathcal{S}} s_i \bmod N \\ &= w^{y+\delta M_S} \mod N \end{aligned}$$

for some $\delta < t$.

CRT Based Function Sharing

The Correction Procedure

- Then tries each $0 \leq j < t$ for

$$(\overline{s}w^{-jM_{\mathcal{S}}})^e \stackrel{?}{\equiv} w \pmod{N}$$

and finds the j_0 satisfying the equality.

- The combiner computes the signature

$$s = \overline{s}w^{-j_0 M_{\mathcal{S}}} \pmod{N}$$

CRT Based Function Sharing

A CRT based 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 a secret $m_0 = \phi(N) = 4p'q'$.

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).

Robustness in CRT Based FSSs

- A FSS is robust if it can withstand active participation of corrupted users in the function evaluation phase.

- The scheme needs to detect wrong partial results.

  1. Each participant creates a proof for the partial result.

  2. These proofs are verified by other participants without knowing the corresponding share.

Robustness

Simple Interactive Proof of Two Discrete Logarithms are Equal

PROVER		VERIFIER
$(x, y) = (g^\alpha, h^\alpha)$		
$r \in_R \mathbb{Z}_q$		
$(a, b) = (g^r, h^r)$	$\overset{a,b}{\rightarrow}$	
	$\overset{c}{\leftarrow}$	$c \in_R \mathbb{Z}_q$
$z \leftarrow r + \alpha c$	$\overset{z}{\rightarrow}$	$g^z \overset{?}{=} ax^c$
		$h^z \overset{?}{=} by^c$

- Chaum and Pedersen, Wallet Databases with Observers, CRYPTO 92

Robustness

for CRT based Threshold RSA Signatures

- In the setup phase

  1. The dealer chooses m_is for Asmuth-Bloom SSS such that $p_i = 2m_i + 1$ is a prime.

  2. He broadcasts $v_i = g_i^{y_i} \bmod p_i$ for each user i where $ord_{p_i}(g_i) = m_i$.

- In the signing phase set

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

  - $v'_i = {v_i}^{M'_{\mathcal{S},i}} \bmod p_i$

  - $w' = w^{M_{\mathcal{S}\setminus\{i\}}} \bmod N$

    and prove that the discrete logs of s_i and v'_i w.r.t. the bases w' and g_i is equal.

Robustness

for Other FSSs

- A similar approach is used for the Paillier's decryption function.

- For sharing ElGamal's encryption function, we need to modify the original scheme slightly.

- The modulus p (N in RSA, N^2 in Paillier) is a prime and $\phi(p)$ is public.

- Solution: use a hidden order version of ElGamal's encryption scheme. (Wei et. al., Cryptographic Primitives based on Groups of Hidden Order, TMMP, 2004)

Conclusions

- Provably secure and practical robust CRT based function sharing schemes exist.

- Not efficient than traditional Shamir based solutions but comparable.

- What about other enhancements or other secret sharing schemes?

Questions