Practical Threshold Signatures: Victor Shoup[1]

Munawar Hasan

February 4, 2020

[1] https://www.shoup.net/papers/thsig.pdf

Initial Setup

- Participants:
  - Players: number of users in the system $[1, \ell]$
  - Dealer: a trusted entity denoted as $\mathcal{D}$, responsible for initial set up
  - Adversary: a non trusted entity denoted as $\mathcal{A}$
- Input parameters to the protocol:
  - k: number of signatures shares needed to obtain a signature
  - t: number of corrupted players

    Note: $k \geq (t+1)$ and $(\ell - t) \geq k$
- Setup
  - $\mathcal{D}$ chooses $p, q \in PRIME$ of bit length L, such that $p = 2 * p' + 1$ and $q = 2 * q' + 1$, hence $\{p', q'\} \in$ Sophie Germain prime
  - $\mathcal{D}$ chooses $e \mid \{e \in PRIME \wedge e > \ell\}$
  - $\mathcal{D}$ computes $n = p * q$; $m = p' * q'$; $\Delta = \ell!$
  - $\mathcal{D}$ computes $d \mid de \equiv 1 \pmod{m}$
  - $\mathcal{D}$ defines a polynomial $f(X)$ of degree $k-1$ in the following way:
    - $f(X) = \sum_{i=0}^{k-1} a_i X^i$, where $a_0 = d$ and $a_i \in \{0, m-1\}$ $\forall i \in \{1, k-1\}$

contd...

- $\mathcal{D}$ computes $s_i = f(i)\ \forall i \in \{1, \ell\}$
- s_i is secret share SK_i of the player i
- $\mathcal{D}$ shares these SK_i with the respective players secretly and securely
- $\mathcal{D}$ chooses $v \in Q_n$, where Q_n is subgroup of squares in Z_n^*
- $\mathcal{D}$ computes $v_i = v^{SK_i}\ \forall i \in \{1, \ell\}$
- $\mathcal{D}$ broadcasts v and all of the v_i
- v_i becomes the verification key of the i^{th} player and v is the verification key of the system
- $\mathcal{D}$ broadcasts public key as $PK = (n, e)$

- At this stage the set up process is complete

Shared Signature

- Let H is a hash function that maps M to Z_n^*, i.e. $H(M) \in Z_n^*$
- Let $x = H(M)$ where M is the message to be signed
- This is a two step process
  - Generating Signature Share:
    - Each player calculates its signature on message M as: $\sigma_i = x^{2*\Delta*SK_i}$
  - Proof of correctness:
    - It is needed to ensure the correctness of the signature shared by the participating players
    - Let $\tilde{x} = x^{4*\Delta}$
    - Let H' be a hash function $\mid H' : \{0,1\}^* \rightarrow \{0,1\}^\gamma$
    - Player i chooses a random number $r \in \{0, 2^{L+2\gamma} - 1\}$ and performs following computations: $v' = v^r$; $x' = \tilde{x}^r$; $c = H'(v, \tilde{x}, VK_i, \sigma_i^2, v', x')$; $z = SK_i * c + r$
    - The proof of correctness is: (z, c)

contd...

- Partial Signature Verification
  - Given (z, c), check if $c \stackrel{?}{=} H'(v, \tilde{x}, VK_i, \sigma_i^2, v^z * VK_i^{-c}, \tilde{x}^z * \sigma_i^{-2c})$
  - Proof:
    - $v' \stackrel{?}{=} v^z * VK_i^{-c}$
      $\Leftrightarrow v^r \stackrel{?}{=} v^{SK_i*c+r} * v^{-SK_i*c}$
    - $x' \stackrel{?}{=} \tilde{x}^z * \sigma_i^{-2c}$
      $\Leftrightarrow \tilde{x}^r = \tilde{x}^{SK_i*c+r} * x^{2*\Delta*SK_i*(-c)}$
      $\Leftrightarrow x^{4*\Delta*r} = x^{4*\Delta*SK_i*c+4*\Delta*r} * x^{-4*\Delta*SK_i*c}$
  - Hence, given (z, c); it can be verified if σ_i was generated by player i or not
- Working on σ_i^2 instead of just σ_i

  $\sigma_i^2 = x^{2*2*\Delta*SK_i} = \tilde{x}^{SK_i}$

  $\Leftrightarrow$ discrete logarithm problem $(v_i = v^{SK_i})$

Combining Signature Shares

- Let $S = \{i_1, ..., i_k\} \subset \{1, .., \ell\}$
- Define: $\lambda_{i,j}^S = \Delta * \frac{\prod_{j' \in S-j}(i-j')}{\prod_{j' \in S-j}(j-j')} \in Z$; Δ insures the evaluation $\in Z$

  $$\Leftrightarrow \Delta * f(0) = \sum_{j \in S} \lambda_{0,j}^S * f(j)$$

  The idea is to compute d from the available signature shares. This can be done in following way:

  Let $w = \sigma_{i_1}^{2*\lambda_{0,i_1}^S} * .... * \sigma_{i_k}^{2*\lambda_{0,i_k}^S}$

  $\Rightarrow w = x^{4\Delta SK_1 * \lambda_{0,i_1}^S} * .... * x^{4\Delta SK_k * \lambda_{0,i_k}^S}$

  $\Rightarrow w = x^{4\Delta * (\lambda_{0,i_1}^S * SK_1 + ... + \lambda_{0,i_k}^S * SK_k)}$

  $\Rightarrow w = x^{4\Delta * \Delta * f(0)} = x^{4\Delta^2 * d}$
- Let $y = w^a * x^b$

  $\Rightarrow y^e = w^{e*a} * x^{e*b}$

  $\Rightarrow y^e = x^{4\Delta^2 * a} * x^{e*b} = x^{e'a+eb}$ where $e' = 4\Delta^2$
- Since $e \in PRIME \wedge e > \ell$, hence $gcd(e', e) = 1$

contd...

$$\Rightarrow y^e = x$$

Toy Example

- Simulate 3 out of 4
- $\ell = 4$; $k = 3$
- $p = 7$; $q = 11$; $p' = 3$; $q' = 5$
- $n = 77$; $m = 15$
- $e = 13$; hence $d = 7$
- $PK = (77, 13)$
- $a_0 = d = 7$; $a_1 = 9$; $a_2 = 6$
- Secret Key: $S_1 = 7$; $S_2 = 4$; $S_3 = 13$; $S_4 = 4$
- Choose $v = 51$; hence $v_1 = 25$; $v_2 = 53$; $v_3 = 4$; $v_4 = 53$;
- Verification Keys: $VK = 51$; $V_1 = 25$; $V_2 = 53$; $V_3 = 4$; $V_4 = 53$;

- Choose an input message $M = hello$; hash function $H = SHA256$
- $x = H(M) = 37; //SHA256(hello) \mod 77$
- $\Delta = 24$
- Generating Signature Shares:
  - $x_1 = 15$; $x_2 = 71$; $x_3 = 36$; $x_4 = 71$;
- Combining Shares:
  - $\lambda_{0,1} = 72$; $\lambda_{0,2} = -72$; $\lambda_{0,3} = 24$
  - $w = 64$
  - $w^e = 36$ and $x^{e'} = 36$
  - Solve for a and b: $e'a + eb = 1$; $a = -4$; $b = 709$
  - $y = w^a x^b = 16$
  - $y^e = (16)^{13} = 37$
- Verify: is $y^e \stackrel{?}{=} x$; $37 \stackrel{?}{=} 37 = SHA256(hello) \mod 77$

References

Victor Shoup
A Computational Introduction to Number Theory and Algebra
Version 2

Victor Shoup
Practical Threshold Signatures
Eurocrypt 2000

İlker Nadi Bozkurt, Kamer Kaya, Ali Aydın Selçuk
Practical Threshold Signatures with Linear Secret Sharing Schemes
Progress in Cryptology – AFRICACRYPT 2009 pp 167-178

The End