ROUGH DRAFT
Verifiably Signed Secret Sharing
A Tool for Identity Escrow

Chris Laas

June 69, 2000

1 Related Work

Verifiable Signature Sharing (VΣS)
Secure distributed computation
Anonymous cash?

2 Model and Definitions

CA, dealer, proxies, reconstructor.

Issuing protocol, sharing protocol, reconstructing protocol. All communications assumed private and unmodifiable in transit.

t-Resilience.

Secret sharing.

3 A Simple Identity Escrow Scheme

Public parameters:

- Large primes p, q such that $q|p-1$
- A generator g with order q in $\mathbb{Z}_p^*$

The CA has a secret key $x \in \mathbb{Z}_q^*$; its public key is $y = g^x \mod p$.

Issuing protocol: a message m may specify attributes of the identity, or may be set to 1 (it may not be zero). The CA signs m with an ElGamal (-like? XXX) signature by choosing a unique value $k \in_{\mathcal{R}} \mathbb{Z}_q^*$, and calculating

$$r = g^k \mod p \tag{1}$$
$$s = k^{-1}(m + xr) \mod q \tag{2}$$

The signature pair (r, s) (along with m if necessary) is sent to $\mathcal{D}$.

Sharing protocol. Summary: the message m and public part of the signature pair r are split into multiplicative shares and distributed. The proxies cooperate to calculate $\rho = r^{e_1 e_2} \mod p$ for randomly chosen e_1 and e_2; $\mathcal{D}$ then proves knowledge of the discrete logarithm of $g^m y^r$ with respect to the base r by answering the challenge of ρ.

1. $\mathcal{D}$ chooses a random value $k_1 \in_{\mathcal{R}} \mathbb{Z}_q^*$, and calculates $r_1 = g^{k_1} \mod p$ and $r_2 = r/r_1 \mod p$. $\mathcal{D}$ sends r_1 to $\mathcal{P}_1$ and sends r_2 to $\mathcal{P}_2$.
2. In the case in which m is not identically 1, $\mathcal{D}$ chooses a random value $j_1 \in_{\mathcal{R}} \mathbb{Z}_q^*$, and calculates $m_1 = g^{j_1} \mod p$ and $m_2 = m/m_1 \mod p$. $\mathcal{D}$ sends m_1 to $\mathcal{P}_1$ and sends m_2 to $\mathcal{P}_2$. (If m is identically 1, $m_1 = m_2 = m = 1$.)
3. $\mathcal{P}_1$ chooses a random value $e_1 \in_{\mathcal{R}} \mathbb{Z}_q^*$, calculates $h = g^{m_1 e_1} \mod p$, $z = y^{r_1 e_1} \mod p$, and $u_1 = r_1^{e_1} \mod p$, and sends h, z, and u_1 to $\mathcal{P}_2$.
4. $\mathcal{P}_2$ chooses a random value $e_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$, calculates $u_2 = r_2^{e_2} \mod p$, and sends u_2 to $\mathcal{P}_1$.
5. $\mathcal{P}_1$ calculates $v_2 = u_2^{e_1} \mod p$ and sends v_2 to $\mathcal{P}_2$.
6. $\mathcal{P}_2$ calculates $\rho = u_1^{e_2} v_2 \mod p$ and sends ρ to $\mathcal{D}$.
7. $\mathcal{D}$ calculates $\gamma = \rho^s \mod p$ and sends γ to $\mathcal{P}_2$.
8. $\mathcal{P}_2$ accepts if and only if $\gamma = (h^{m_1} z^{r_2})^{e_2} \pmod p$.

Proposition 1: If $\mathcal{D}$ can fool $\mathcal{P}_2$ into accepting despite not having been issued an (r, s) pair by the $\mathcal{CA}$, then $\mathcal{D}$ can forge an ElGamal signature on the message m, or can break the zero-knowledge proof of knowledge of a discrete logarithm used here.

Proposition 2: If $\mathcal{D}$ and $\mathcal{P}_2$ follow the protocol, $\mathcal{P}_1$ learns no value correlated with r or m from its view of the protocol. (Information-theoretic.)

Speculation 3: If $\mathcal{D}$ and $\mathcal{P}_1$ follow the protocol, $\mathcal{P}_2$ learns no value correlated with r or m from its view of the protocol. (Not sure. Heuristic.)

Reconstruction protocol: $\mathcal{P}_1$ and $\mathcal{P}_2$ send their values r_1, m_1, r_2, and m_2 to $\mathcal{R}$. $\mathcal{R}$ calculates $r = r_1 r_2$ and $m = m_1 m_2$.

References

Verifiable Signature Sharing (EUROCRYPT '95)