The Theory Behind Blockchains (Spring 19)
Recitation 9

Eliad Tsfadia

1 Unique Signature

Recall that in recitation 2 we constructed a signature-scheme using a trapdoor-permutation (TDP) triplet (Gen_T, F, F^{-1}) and a random oracle h as follows:

1. $\text{Gen}(1^n)$: Sample $(pk, sk) \leftarrow Gen_T(1^n)$.
2. $\text{Sign}_{sk}(m)$: Output $\sigma = F_{sk}^{-1}(h(m))$.
3. $\text{Vrfy}_{pk}(m, \sigma)$: Output $1 \iff F_{pk}(\sigma) = h(m)$.

By definition, since F_{sk} is a permutation, then F_{sk}^{-1} is also a permutation which yields that each signature σ on a message m is uniquely defined.

But how we construct a TDP?

2 TDP from RSA

Recall that we also mentioned in recitation 2 that we can implement TDP from RSA: $\text{Gen}_{RSA}(1^n)$ chooses n-bit length primes p and q, sets $N = p \cdot q$, chooses an integer e which is relatively prime to $\phi(N)$, computes $d = e^{-1} \bmod \phi(N)$ and returns $(pk = (N, e), sk = d)$ (recall that $\phi(N) = |\mathbb{Z}_N^*| = (p-1)(q-1)$, where $\mathbb{Z}_N^*$ is the set of all integers in $[N-1]$ that are relatively prime to N). For any $x \in [N-1]$, $F_{pk}(x) = x^e \bmod N$, and for any $y \in [N-1]$, $F_{sk}^{-1}(y) = y^d \bmod N$.

Claim 1 (Correctness)**.** *For any $x \in [N-1]$, $F_{sk}^{-1}(F_{pk}(x)) = x$.*

Proof Sketch. The set $\mathbb{Z}_N^*$ is a group of size $\phi(N)$ (with respect to multiplications modulo N). Therefore, if $x \in \mathbb{Z}_N^*$, then $x^{\phi(N)} = 1 \bmod N$. By construction, $e \cdot d = 1 \bmod \phi(N)$. Hence,

$$F_{sk}^{-1}(F_{pk}(x)) = x^{e \cdot d} \bmod N = x^{k \cdot \phi(N)+1} \bmod N = x.$$

If $x \notin \mathbb{Z}_N^*$ then $gcd(x, N) = p$ or $gcd(x, N) = q$. Assume $gcd(x, N) = p$, i.e. $x = kp$ for some $k > 0$. Then it holds that

$$x^{ed} = 0 \bmod p = x \bmod p \tag{1}$$

Since $gcd(x, q) = 1$, (i.e., x can be seen as an element in the group $\mathbb{Z}_q^* = \{1, \ldots, q-1\}$) then $x^{\phi(q)} = x^{q-1} = 1 \bmod q$. Hence

$$x^{ed} = x^{ed-1} \cdot x = x^{\ell(p-1)(q-1)} \cdot x = (x^{q-1})^{\ell(p-1)} \cdot x = x \bmod q \tag{2}$$

We know use Chinese Remainder Theorem which states that for all x and y:

$$x = y \bmod p \wedge x = y \bmod q \implies x = y \bmod pq$$

By applying it on (1) and (2), we conclude that $F_{sk}^{-1}(F_{pk}(x)) = x^{e \cdot d} \bmod N = x \bmod N$. □

3 Verifying that (N, e) were chosen correctly

A crucial assumption that made the above construction to be TDP is the fact that e is invertible modulo $\phi(N)$. If, for instance, $\phi(N) = k \cdot e$ for some $k > 0$, then the transformation $x \mapsto x^e \bmod N$ is not invertible (actually, it is an e-to-1 mapping):

$$x^e \bmod N = (x^{k+1})^e \bmod N = (x^{2k+1})^e \bmod N = \ldots = (x^{(e-1)k+1})^e \bmod N$$

Therefore, if a malicious party chooses bad values of N and e, it can create for itself a non-unique signature without being detected and this can give it an advantage in Algorand's system.

We need that each player will generate a proof that it's N, e indeed were chosen correctly. We do so by describing an interactive public-coin protocol between a prover P (which holds (N, e), d and also know p and q) and a verifier V (which only knows (N, e)), and this protocol can be translate into a non-interactive proof using Fiat-Shamir (see recitation 8 for more details).

The protocol is very simple: The verifier chooses a random $x \in \mathbb{Z}_N^*$ and sends $x^e \bmod N$ to the prover P. The prover then sends back $x' \in \mathbb{Z}_N^*$ to V and V checks whether $x = x'$. Note that if the prover chooses N, e such that e divides $\phi(N)$, then as we claimed before, the mapping $x \mapsto x^e$ is e-to-1 mapping, which means that the prover can guess correctly and cheat only with probability $1/e$. In general, if $gcd(\phi(N), e) = \ell \geq 2$, then the mapping $x \mapsto x^e$ is ℓ-to-1 (HW) and the prover can only cheat with probability $1/\ell$. The protocol is repeated $\omega(\log n)$ times (sequentially or in parallel) and a cheating prover will be able to cheat only with probability $(1/\ell)^{\omega(\log n)} = \text{negl}(n)$.