CS181A 11: Miscellany

Digital Signatures

We describe a simple application of RSA for creating digital signatures. In a signature scheme, Alice would like to send a signed message to Bob. The message is signed so that Bob can be confident that the sender of the message is indeed Alice (and not Eve).

> Assume that both Alice and Bob have their own RSA public and secret keys. We denote Alice's and Bob's public keys as e_A, e_B whereas their secret keys are d_A, d_B. When Alice needs to send a message m to Bob, she "decrypts" her message using her secret key d_A to get $m' = D_A(m)$. Then, she "encrypts" this using Bob's public keys and obtains $c = E_B(m')$. Bob recovers the *signed* message m' by decrypting c using $m' = D_B(c)$. To verify that the signed message is from Alice, Bob computes "encrypts" m' using Alice's public keys to get $m = E_A(m')$.

Hash functions

An alternative way of implementing a digital signature is to use a cryptographic hash function h which creates a signature $h(m)$ for a message m. The basic requirement on h is that it must be intractable to find a different m' with the same signature as m; that is, $m \neq m'$ but $h(m) = h(m')$. This is called *existential forgery*.

We describe a simple and natural hash function based on the hardness of the Discrete Logarithm problem. This construction is due to Chaum, van Heijst and Pfitzmann.

> Let $p = 2q + 1$ be a safe prime, where both p and q are primes. Suppose α and β are generators for $\mathbb{Z}_p^\star$. Define the hash function $H : \mathbb{Z}_q \times \mathbb{Z}_q \to \mathbb{Z}_p^\star$:
>
> $$H(x, y) = \alpha^x \beta^y \bmod p.$$

Claim 1. *If there is an efficient algorithm to find collision for H, then there is an efficient algorithm to compute $\log_\alpha \beta$ (the discrete logarithm base α of β).*

Proof. Suppose $H(x_1, x_2) = H(x_3, x_4)$, where $(x_1, x_2) \neq (x_3, x_4)$. Let $d = \gcd(x_4 - x_2, p-1)$. Note $d \in \{1, 2, q, p-1\}$.

1. Case $d = 1$:
   So, there is an integer y so $(x_4 - x_2)y \equiv 1 \pmod{p-1}$. Since $H(x_1, x_2) = H(x_3, x_4)$, we have
   $$\alpha^{x_1 - x_3} \equiv \beta^{x_4 - x_2} \pmod{p}.$$
   Thus, $\alpha^{(x_1 - x_3)y} \equiv \beta \pmod{p}$. This implies $\log_\alpha \beta = (x_1 - x_3)y$.

2. Case $d = 2$:
   We must have $\gcd(x_4 - x_2, q) = 1$. Thus, there is an integer y so $(x_4 - x_2)y = kq + 1$, for some integer k. Then,
   $$\alpha^{(x_1 - x_3)y} = \beta^{(x_4 - x_2)y} = \beta^{kq}\beta = (-1)^k \beta = \pm\beta \pmod{p}.$$
   So, either $(x_1 - x_3)y$ or $q + (x_1 - x_3)y$ yields $\log_\alpha \beta$, since $\alpha^q \equiv -1 \pmod{p}$.

3. Case $d = q$:
   Note $-(q-1) \leq x_4 - x_2 \leq q - 1$ since $x_2, x_4 \in \mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. Thus, $d = q$ is impossible. If $x_4 - x_2$, then $d = 2q$ instead of $d = q$.

4. Case $d = p - 1 = 2q$:
   Since $-(q-1) \leq x_4 - x_2 \leq q - 1$, we must have since $x_4 - x_2 = 0$. This implies $x_1 = x_3$, which contradicts the assumption that $(x_1, x_2) \neq (x_3, x_4)$.

□

Birthday attack

Suppose we randomly (and uniformly) toss m balls into n bins. What is the probability that there is a bin which contains more than one ball? It is easier to work with the complementary event $\mathcal{Z}$ which is that all n bins contain at most one ball. Let $\mathcal{A}_j$ be the event that the first j balls are tossed into distinct bins and let $\mathcal{B}_j$ be the event that the j-th ball is tossed into an empty bin. Then,

$$\mathbb{P}[\mathcal{A}_m] = \prod_{j=1}^{m} \mathbb{P}[\mathcal{B}_j | \mathcal{A}_{j-1}] = \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right) \leq \exp\left(-\frac{1}{n}\sum_{j=1}^{m-1} j\right),$$

since $1 - x \leq e^{-x}$. Thus, the probability that no bin contains more than one ball is at most $\exp(-m^2/n)$. This probability is a constant (bounded away from zero) if $m \geq \Omega(\sqrt{n})$.

References

[1] D. Chaum, E. Van Heijst and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer," in Proc. of *Advances in Cryptology* (CRYPTO91), Lecture Notes in Computer Science **576**, 470-484, Springer-Verlag, 1992.