Today: Digital Signatures

- Security definition
- Hash & Sign & RSA Full Domain Hash.
- Schnorr signature scheme
  - Schnorr identification (ID) scheme
  - Fiat-Shamir paradigm for converting ID schemes into signature schemes.

Recall:

Def A digital signature consists of PPT algorithms $(Gen, Sign, Ver)$ with the following syntax:

- $Gen(1^\lambda)$: Generates a pair (PK, SK) of public key & secret key.
- $Sign(SK, m)$: Outputs a signature σ.
- $Ver(PK, m, \sigma)$: Outputs 0/1.

Correctness: $\forall m \in \mathcal{M} \quad \forall \lambda \in \mathbb{N}$.

$$\Pr_{(PK,SK) \leftarrow Gen(1^\lambda)}[\, Ver(PK, m, Sign(SK, m)) = 1 \,] = 1$$

Security

Def: A signature scheme (Gen, Sign, Ver) is secure against adaptive chosen msg attacks if $\forall$ PPT A

existentially unforgable

$$\Pr\left[A^{\text{Sign}(SK,\cdot)}(PK) = (m^*, \sigma^*) : \begin{array}{l}\text{Ver}(PK, m^*, \sigma^*) = 1 \\ \& \\ m^* \text{ was not queried}\end{array}\right] = \text{negl}(\lambda)$$

$(PK, SK) \leftarrow \text{Gen}(1^\lambda)$

Last lecture: Textbook RSA signature scheme.

$\text{Gen}(1^\lambda)$: Generate random primes $p, q \in \{0,1\}^\lambda$.

Compute $n = p \cdot q$

Choose $e \in \mathbb{Z}^*_{\phi(n)}$ (common choice $e=3$)

Compute $d = e^{-1} \bmod \phi(n)$ using extended GCD.

Output $PK = (n, e)$, $SK = (n, d)$

$\text{Sign}((n,d), m)$: Output $\sigma = m^d \bmod n$

$\text{Ver}((n,e), m, \sigma)$: Output 1 iff $\sigma^e = m \bmod n$.

Note: This scheme is not secure against adaptive chosen msg attacks (ACMA).

Eg. Easy to sign the msg $m = \sigma^e \bmod n$, for any σ.

Secure version: Hash & Sign

(also known as: Full Domain Hash)

Rather than signing m sign $H(m)$, where H is a hash function.

$Gen(1^\lambda)$: same as textbook RSA: $PK=(n,e)$, $SK=(n,d)$ s.t. $e \cdot d = 1 \bmod \varphi(n)$

$Sign(SK,m)$: $H(m)^d \bmod n$

$Verify(PK,m,\sigma)$ outputs 1 iff $\sigma^e = H(m) \bmod n$.

Claim: This scheme is secure against ACMA in the Random Oracle Model (ROM), under RSA Assumption

$n,e,x \xrightarrow{HARD} x^d \bmod n$

Pf idea: First note that without signing oracle it is hard to generate any valid signature, since it requires breaking RSA in ROM.

Next notice that signing oracle is useless since it can be efficiently simulated together with the random oracle, as follows:

Given msg m, choose $\sigma \leftarrow \mathbb{Z}_n^*$, compute $r = \sigma^e \bmod n$, set $H(m) := r$, and output σ as a signature for m.

Whenever m is sent to H answer with a random r computed by $r = \sigma^e \bmod n$ for random $\sigma \leftarrow \mathbb{Z}_n^*$.

Note: Hash & Sign also used for efficiency purposes! (Sig length depends on hash, not msg)

Schnorr Signature Scheme (Discrete Log based)

The Schnorr signature scheme can be thought of as an identification scheme, on which we apply the Fiat-Shamir paradigm

Schnorr Identification Scheme

An identification scheme is a protocol where a user proves her identity (ownership of a public key).

Schnorr's ID scheme is based on Discrete log:

Let G be a group of prime order q, and let g be a generator of G.

Ex. Choose a random large prime q and a random r

s.t. $p = q \cdot r + 1$ is prime.

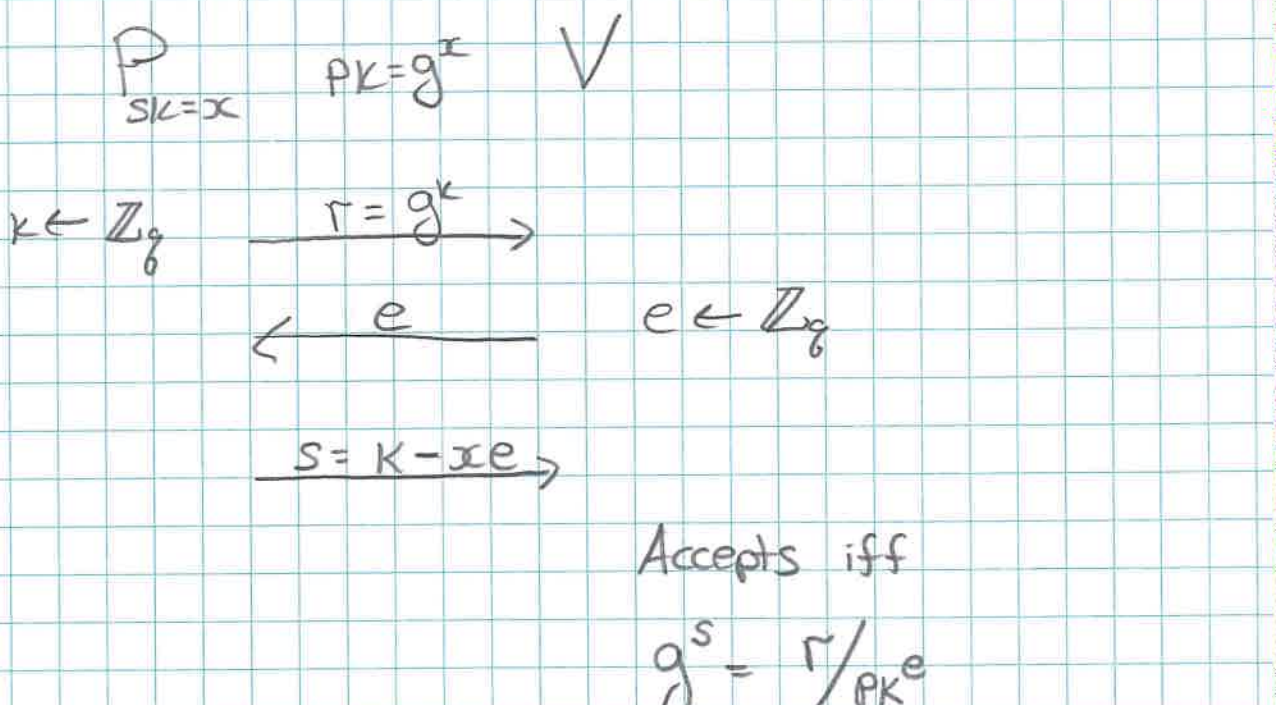
Note that $|\mathbb{Z}_p^*| = p - 1 = q \cdot r$

Let $G = \{ h^r \bmod p \;\; h \in \mathbb{Z}_p^* \} \subseteq \mathbb{Z}_p^*$

To choose a generator $g \in G$, choose any $h \in \mathbb{Z}_p^*$ s.t. $h^r \neq 1 \bmod p$.

Typically, p large enough to prevent discrete log attacks in $\mathbb{Z}_p^*$; q can be smaller, but large enough to resist birthday attacks. (Eg. $p \in \{0,1\}^{1024}$, $q \in \{0,1\}^{160}$)

Schnorr's ID scheme assumes user have key pairs of the form $(PK, SK) = (g^x, x)$ for $x \leftarrow \mathbb{Z}_q$ (Similar to El-Gamal)

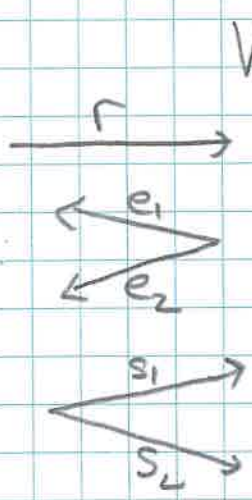


Note: If P convinces V to accept (with non-negl prob) then P "must know" the secret key

$\Downarrow$

$\equiv$ we can use P to efficiently find the secret key.

Idea: "Rewind" P: Run P on 2 different challenges e_1, e_2



If both executions accept then:

$$g^{s_1} PK^{e_1} = g^{s_2} PK^{e_2} = r$$

$\Downarrow$

$$g^{\frac{s_1 - s_2}{e_2 - e_1} \bmod q} = PK$$

$\Rightarrow \frac{s_1 - s_2}{e_2 - e_1} \bmod q$ is the secret key corresponding to PK.

ID scheme is required to have secrecy guarantees:

Security against passive impersonation attacks:

It is hard to impersonate even after observing

poly many protocol executions.

Claim: Impersonator gains no information by eavesdropping.

(PK, r, e, s) can be generated efficiently (w.o. knowing SK)

given only PK as follows:

Choose random $e, s \leftarrow \mathbb{Z}_q$ and let

$r = g^s PK^e$.

This is called: Honest Verifier Zero Knowledge.

Fiat-Shamir Paradigm

A heuristic for converting ID schemes into signature schemes

$Sign(SK, m) = (r, e, s)$ where $e = H(m, r)$

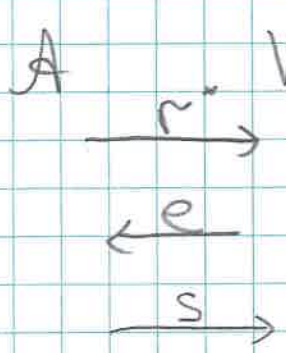
$Ver(PK, m, \underbrace{\sigma}_{(r,e,s)})$ outputs 1 iff verifier of ID scheme accepts $(PK, (r, e, s))$ and $e = H(m, r)$

Claim: ∀ ID scheme that is secure against passive impersonation attacks, the signature scheme resulting by applying the Fiat-Shamir paradigm on ID, is secure against ACMA in the ROM.

Pf idea: Suppose $\exists$ PPT $A^{Sign(SK,\cdot)}(PK)$ that generates valid msg/signature pair.

We will use A to attack the ID scheme.

A outputs a valid signature (r^*, e^*, s^*) where $e^* = H(m, r^*)$



A can generate valid s for non-negl fraction of oracle answers $e = H(m, r)$

Intuitively, interacting w. an oracle H is no different than interacting w. the verifier.

Oracle access to Sign(SK, ·) (in ROM) is similar to passive eavesdropping.

This scheme is very similar to Digital Signature Standard (DSS).

Main difference: $r = g^k \bmod p \bmod q$ ← for efficiency reasons.

$g \in \mathbb{Z}_p^*$ $|\langle g \rangle| = q$.

Also in DSS $e = H(m)$ as opposed to $e = H(m, r)$.

The version where $e = H(m)$ is not known to be secure in the ROM.

[it is insecure w. Schnorr but not known to be insecure in DSS]