Cryptography
Lecture 22

Announcements

- HW5 due on 4/24 at11:59pm
- HW6 posted, due on 5/8 at 11:59pm
- No lecture (or quizzes) on 4/24 and 4/29
    - Please watch videos linked in the Canvas announcement

Agenda

- Last time:
  - Diffie-Hellman Key Exchange
  - Public Key Encryption
  - ElGamal (11.5)

- This time:
  - Digital Signatures Definitions (13.2)
  - DL-Based Signatures (13.5)

Digital Signatures Definition

A digital signature scheme consists of three ppt algorithms $(Gen, Sign, Vrfy)$ such that:

1. The key-generation algorithm Gen takes as input a security parameter 1^n and outputs a pair of keys (pk, sk). We assume that pk, sk each have length at least n, and that n can be determined from pk or sk.
2. The signing algorithm $Sign$ takes as input a private key sk and a message m from some message space (that may depend on pk). It outputs a signature σ, and we write this as $\sigma \leftarrow Sign_{sk}(m)$.
3. The deterministic verification algorithm $Vrfy$ takes as input a public key pk, a message m, and a signature σ. It outputs a bit b, with $b = 1$ meaning valid and $b = 0$ meaning invalid. We write this as $b \coloneqq Vrfy_{pk}(m, \sigma)$.

Correctness: It is required that except with negligible probability over (pk, sk) output by $Gen(1^n)$, it holds that $Vrfy_{pk}\big(m, Sign_{sk}(m)\big) = 1$ for every message m.

Digital Signatures Definition: Security

Experiment $SigForge_{A,\Pi}(n)$:

1. $Gen(1^n)$ is run to obtain keys (pk, sk).
2. Adversary A is given pk and access to an oracle $Sign_{sk}(\cdot)$. The adversary then outputs (m, σ). Let Q denote the set of all queries that A asked to its oracle.
3. A succeeds if and only if
   1. $Vrfy_{pk}(m, \sigma) = 1$
   2. $m \notin Q$.

In this case the output of the experiment is defined to be 1.

Definition: A signature scheme $\Pi = (Gen, Sign, Vrfy)$ is existentially unforgeable under an adaptive chosen-message attack, if for all ppt adversaries A, there is a negligible function neg such that:

$$\Pr\left[SigForge_{A,Pi}(n) = 1\right] \leq neg(n).$$

Overview of DL-based Signatures

- Discrete-Log-based signatures can be implemented using Elliptic Curves.
  - They are therefore more efficient than RSA-based signatures (signatures are far smaller).
- DL-based are preferred in Bitcoin
- Bitcoin currently uses ECDSA = Elliptic Curve Digital Signature Algorithm
- We will be learning about Schnorr signatures.
- Similar to ECDSA but have some better properties.
- ~~Many proponents of switching Bitcoin signatures to Schnorr signatures.~~

Implemented in Bitcoin in November 2021 within the Taproot upgrade as an alternative to the Elliptic Curve Digital Signature Algorithm (ECDSA)

Outline

- We will first construct an **Identification Scheme**
  - A way to prove knowledge of a secret key corresponding to a public key without revealing the secret key
  - Provides a form of "zero knowledge"
  - E.g. public key = g^x, secret key = x.
  - Prove that I know x, without revealing what x is
  - If I reveal x, someone can impersonate me next time.
- Use the **Fiat-Shamir transform** to convert an **Identification Scheme** into a **Signature Scheme**.

Identification Schemes

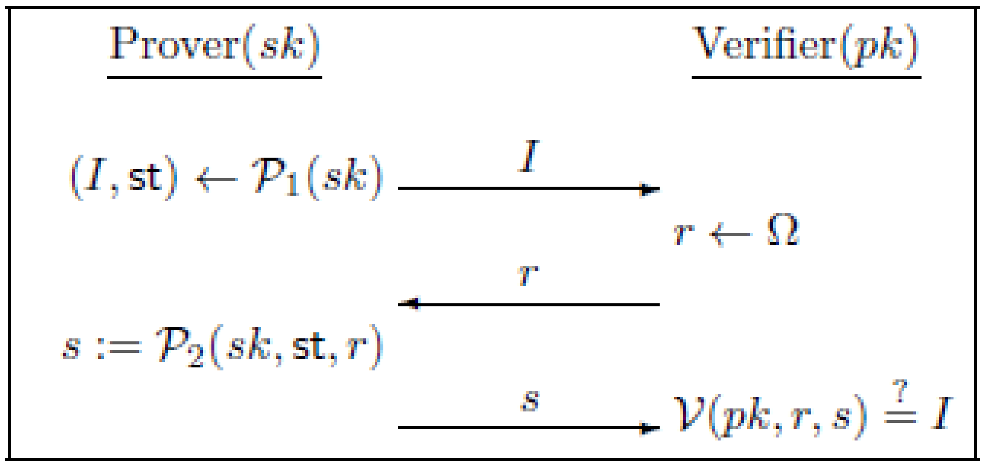

FIGURE 12.1: A 3-round identification scheme.

Identification Schemes

The identification experiment $\mathsf{Ident}_{\mathcal{A},\Pi}(n)$**:**

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys* (pk, sk).
2. *Adversary* $\mathcal{A}$ *is given* pk *and access to an oracle* $\mathsf{Trans}_{sk}(\cdot)$ *that it can query as often as it likes.*
3. *At any point during the experiment,* $\mathcal{A}$ *outputs a message* I. *A uniform challenge* $r \in \Omega_{pk}$ *is chosen and given to* $\mathcal{A}$, *who responds with* s. *(We allow* $\mathcal{A}$ *to continue querying* $\mathsf{Trans}_{sk}(\cdot)$ *even after receiving c.)*
4. *The experiment evaluates to 1 if and only if* $\mathcal{V}(pk, r, s) \stackrel{?}{=} I$.

DEFINITION 12.8 *Identification scheme* $\Pi = (\mathsf{Gen}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ *is* secure against a passive attack, *or just* secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there is a negligible function* negl *such that:*

$$\Pr[\mathsf{Ident}_{\mathcal{A},\Pi}(n) = 1] \le \mathsf{negl}(n).$$

The Schnorr Identification Scheme

Prover(x)		Verifier($\mathbb{G}, q, g, y$)
$k \leftarrow \mathbb{Z}_q$		
$I := g^k$	$\xrightarrow{\quad I \quad}$	
		$r \leftarrow \mathbb{Z}_q$
	$\xleftarrow{\quad r \quad}$	
$s := [rx + k \bmod q]$		
	$\xrightarrow{\quad s \quad}$	check whether $g^s \cdot y^{-r} \stackrel{?}{=} I$

FIGURE 12.2: An execution of the Schnorr identification scheme.

Security Analysis

Theorem: If the Dlog problem is hard relative to G then the Schnorr identification scheme is secure.

Security Analysis

Idea of proof:

- Oracle can generate correctly distributed transcripts without knowing x.
  - How?

Security Analysis

Idea of proof:

- Given an attacker A who successfully responds to challenges with non-negligible probability, can construct an attacker A' who extracts the discrete log x of y by **rewinding**.

From Identification Schemes to Signatures: The Fiat-Shamir Transform

CONSTRUCTION 12.9

Let $(\mathsf{Gen}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ be an identification scheme, and construct a signature scheme as follows:

- Gen: on input 1^n, simply run $\mathsf{Gen}(1^n)$ to obtain keys pk, sk.

  The public key pk specifies a set of challenges Ω_{pk}. As part of key generation, a function $H : \{0,1\}^* \to \Omega_{pk}$ is specified, but we leave this implicit.

- Sign: on input a private key sk and a message $m \in \{0,1\}^*$, do:

  1. Compute $(I, \mathsf{st}) \leftarrow \mathcal{P}_1(sk)$.
  2. Compute $r := H(I, m)$.
  3. Compute $s := \mathcal{P}_2(sk, \mathsf{st}, c)$

  Output the signature (r, s).

- Vrfy: on input a public key pk, a message m, and a signature (r, s), compute $I := \mathcal{V}(pk, r, s)$ and output 1 if and only if $H(I, m) \stackrel{?}{=} r$.

The Fiat-Shamir transform.

Security Analysis

Theorem: Let Π be an identification scheme, and let Π' be the signature scheme that results by applying the Fiat-Shamir transform to it. If Π is secure and H is modeled as a random oracle, then Π' is secure.

The Schnorr Signature Scheme

CONSTRUCTION 12.12

Let $\mathcal{G}$ be as described in the text.

- Gen: run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Choose uniform $x \in \mathbb{Z}_q$ and set $y := g^x$. The private key is x and the public key is $(\mathbb{G}, q, g, y)$. As part of key generation, a function $H : \{0,1\}^* \to \mathbb{Z}_q$ is specified, but we leave this implicit.
- Sign: on input a private key x and a message $m \in \{0,1\}^*$, choose uniform $k \in \mathbb{Z}_q$ and set $I := g^k$. Then compute $r := H(I, m)$, followed by $s := [rx + K \bmod q]$. Output the signature (r, s).
- Vrfy: on input a public key $(\mathbb{G}, q, g, y)$, a message m, and a signature (r, s), compute $I := g^s \cdot y^{-r}$ and output 1 if $H(I, m) \stackrel{?}{=} r$.

The Schnorr signature scheme.