Introduction to Cryptology

Lecture 26

Announcements

- HW 10 and Scholarly Paper EC due today
- Final Exam Info:
  – Thursday, 5/17 from 1:30-3:30pm in CSI 1122 (our regular classroom)
  – Final review sheet on course webpage, solutions are on Canvas
  – Cheat sheet for final will be posted
  – TA OH 5/10 from 5-6pm
  – Instructor OH 5/15 from 3-4:30pm.

Agenda

- Last time:
  - Digital Signatures Definitions (12.2-12.3)
  - RSA Signatures (12.4)

- This time:
  - Dlog-based signatures (12.5)

  ****We did not cover this due to time, but I am posting it for those who may be interested*****
  - Certificates and PKI, TLS/SSL (12.7-12.8)

Identification Schemes

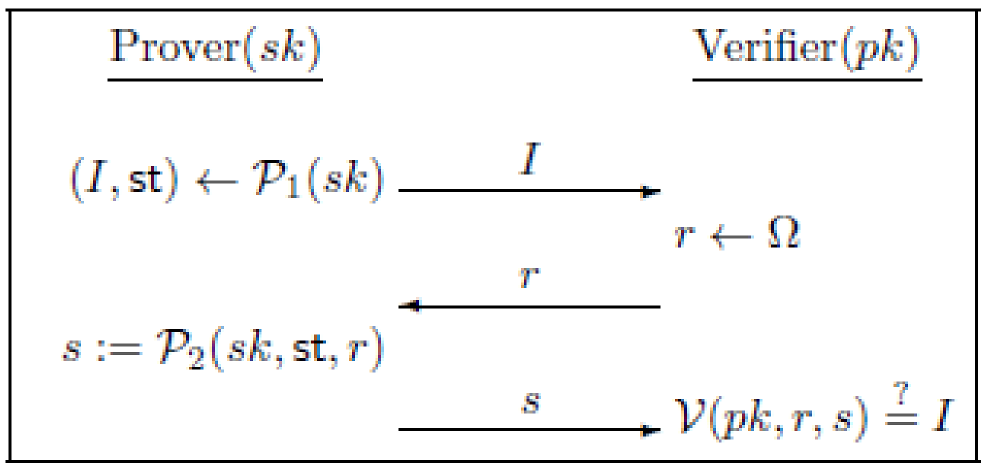

FIGURE 12.1: A 3-round identification scheme.

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:*

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

The Schnorr Identification Scheme

$\underline{\text{Prover}(x)}$		$\underline{\text{Verifier}(\mathbb{G}, q, g, y)}$
$k \leftarrow \mathbb{Z}_q$		
$I := g^k$	$\xrightarrow{\quad I \quad}$	
		$r \leftarrow \mathbb{Z}_q$
	$\xleftarrow{\quad r \quad}$	
$s := [rx + k \bmod q]$		
	$\xrightarrow{\quad s \quad}$	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 $H : \{0,1\}^* \rightarrow \Omega_{pk}$ 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.

Certificates and Public-Key Infrastructure

A single certificate authority

- pk_{CA} must be distributed over an authenticated channel
  - Need only be carried out once
- Usually, pk_{CA} included in browser, browser programmed to automatically verify certificates as they arrive.
- To obtain certificate, must prove that url is legitimate.
- All parties must completely trust CA.

Multiple certificate authorities

- Parties can choose which CA to use to obtain a certificate.
- Parties can choose which CA’s certificates to trust.
- Problem: some CA may become compromised.
- Each user must manually decide which CA to trust.

Delegation and certificate chains

- Example of certificate chain:

$$pk_A, cert_{B \to A}, pk_B, cert_{C \to B}$$

Need only trust Charlie in the above example.

- Certificate asserts that legitimate party holds public key and *that the party is trusted to issue other certificates.*
  - Delegation of CA's ability to issue certificates

The "web of trust" model

- Model is used by PGP ("pretty good privacy") email encryption software for distribution of public keys.
- Anyone can issue certificates to anyone else
- Each user must decide who to trust
- Example:
  – Alice holds pk_1, pk_2, pk_3 for users C_1, C_2, C_3
  – Bob has certificates $cert_{C_1 \to B}, cert_{C_3 \to B}, cert_{C_4 \to B}$
- Public keys and certificates can be stored in a central database.

Invalidating Certificates

- Expiration: Include expiration date as part of the certificate.
  - Very coarse grained method. E.g. employee leaves company but certificate does not expire for a year.
- Revocation
  - CA includes a serial number in every certiciate it issues.
  - At the end of each day, the CA will generate a certificate revocation list (CRL) with the serial numbers of all revoked certificates.
  - CA will sign the CRL and the current date.
  - Signed CRL is then widely distributed.

Putting it all together: SSL/TLS

- TLS: Transport Layer Security Protocol
  - Protocol used by browser when connecting via https
- Standardized protocol based on a precursor called SSL (Secure Socket Layer).
  - Latest SSL version: SSL 3.0
  - TLS version 1.0 released in 1999
  - TLS version 1.1 in 2006
  - TLS version 1.2 (current) in 2008
  - 50% of browsers still use TLS 1.0
- Allows a client (web browser) and a server (website) to agree on a set of shared keys and then use those keys to encrypt and authenticate their subsequent communication.
- Two parts:
  - Handshake protocol performs authenticated key exchange to establish the shared keys
  - Record-layer protocol uses shared keys to encrypt/authenticated the communication.
- Typically used for authentication of servers to clients (usually only servers—websites—have certificates).