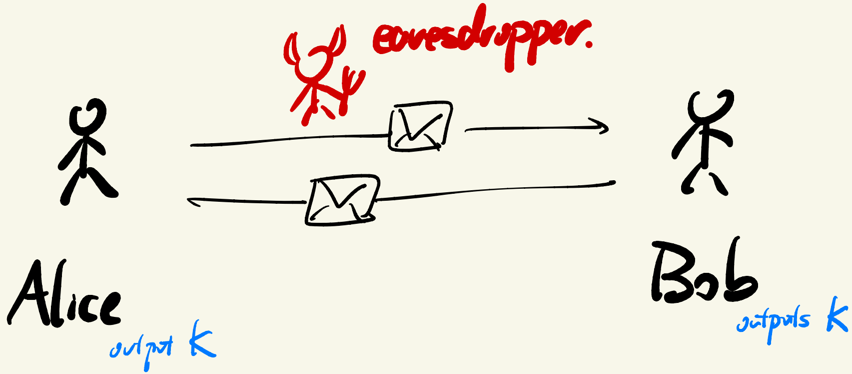
Lec 8. Public-key cryptography



Key Agreement (Problem/Protocol)

- a pair of interactive algorithms A, B

Game

- $\langle A(1^\lambda), B(1^\lambda) \rangle$
- passive adversary.
  - weak ① hard to guess k $\xrightarrow{\text{def}}$ $\mathcal{A}$ is given the transcript, guess k'
    $\mathcal{A}$ win iff $k' = k$
  - strong ② k "looks" random $\xrightarrow{\text{def}}$ $\mathcal{A}$ cannot distinguish (transcript, k), (transcript, a fresh key)

trapdoor one-way function.

$Gen(1^\lambda) \to pp$ (public parameter), tp (trapdoor)

$f(pp, x) = f_{pp}(x)$, f_{pp} is injective

$f_{pp}: X_{pp} \to Y_{pp}$

$Invert(pp, tp, y) \to x$ s.t. $f_{pp}(x) = y$

one-wayness: $\forall$ p.p.t. A

$$\Pr_{pp, x}[A(pp, f_{pp}(x)) \to x] \leq negl(\lambda)$$

notation

$Gen(1^\lambda) \to f, tp$

$f(x)$

$Invert(tp, f(x)) = x$

Key-Agreement based on trapdoor OWF

$A(1^\lambda)$ | B

$Gen \to f, tp$ $\xrightarrow{f}$ sample $k \xleftarrow{\$} ??$

$Invert(tp, f(k)) \to k$ $\xleftarrow{f(k)}$

Constructions of trapdoor one-way function

key agreement based on RSA

A

sample λ-bit primes p, q

$N = pq \quad \phi(N) = (p-1)(q-1)$

sample e, d s.t. $ed \equiv 1 \mod \phi(N)$

$\xrightarrow{N, e}$

B

Sample random $x \leftarrow \mathbb{Z}_N^*$

$\xleftarrow{y = x^e}$

Compute $(x^e)^d = x^{ed} = x$

Output x

Output x

Trapdoor OWF based RSA

$Gen(1^\lambda) \to N = pq$

e, d s.t. $ed \equiv 1 \mod \phi(N)$

$pp = (N, e) \quad tp = d$

$f_{pp}(x) = x^e$

$f_{pp} : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$

Assumptions

- factorization is hard.

Game

random λ-bit primes p, q

$A(N = pq) \to (p', q')$

A wins iff $\{p', q'\} = \{p, q\}$

$\forall$ p.p.t. A.

A wins with negl probability.

- RSA assumption

Game

random λ-bit prime p, q

$N = pq$

random e s.t. $\gcd(e, \phi(N)) = 1$

random x

$A(N, e, x^e) \to x'$

A wins iff $x' = x$

$\forall$ p.p.t. A.

A wins with negl probability.

Self reduction

Assume A. construct A' $\forall N, e$

A wins w.p. $\frac{1}{poly(\lambda)}$

conditioning on N, e

$\Rightarrow A'$ wins w.p. $1 - negl(\lambda)$

condition N, e

$A'(N, e, x^e)$

sample z. $A(N, e, (xz)^e)$

w.p. $\frac{1}{poly(\lambda)}$ A output $x \cdot z$

$N = pq \quad \phi(N) = (p-1)(q-1)$

$\mathbb{Z}_N^* = \{1 \le i \le N \mid \gcd(i, N) = 1\}$

$|\mathbb{Z}_N^*| = \phi(N) = (p-1)(q-1)$

Sample e, d, $\quad e \cdot d \equiv 1 \mod \phi(N)$

$a^{ed} = a \mod N$

- strong RSA assumption

Game

random λ-bit prime p, q

$N = pq$

~~random e s.t. $\gcd(e, \phi(N)) = 1$~~

~~random x~~ random y

$A(N, y) \to (x, e)$

A wins iff $x^e = y$

$\forall$ p.p.t. A,

A wins with negl probability.

- RSA assumption for fixed odd e

Game

random λ-bit prime p, q

$\gcd(e, p-1) = \gcd(e, q-1) = 1$

random x

$A(N, e, x^e) \to x'$

A wins iff $x' = x$

$\forall$ p.p.t. A,

A wins with negl probability.

- Another RSA assumption

Game

random λ-bit safe primes p, q

$p = 2p' + 1$

$q = 2q' + 1$

p', q' are primes

random x

$A(N, e, x^e) \to x'$

A wins iff $x' = x$

$\forall$ p.p.t. A,

A wins with negl probability.

Computational Diffie-Hellman Assumption

$Gen(1^\lambda) \to G, g$

$\forall$ p.p.t. A

$x, y \xleftarrow{\$} \{0, 1, \ldots, |G|-1\}$

$$\Pr[A(G, g, g^x, g^y) = g^{xy}] \leq negl(\lambda)$$

Discrete log assumption

$Gen(1^\lambda) \to G, g$

$\forall$ p.p.t. A

$x \xleftarrow{\$} \{0, 1, \ldots, |G|-1\}$

$$\Pr[A(G, g, g^x) = x] \leq negl(\lambda)$$

$order(a)$: $a^{order(a)} = 1$; $\forall 1 \leq t < order(a): a^t \neq 1$

G, g — G: group, g: a generator

$G' = \{1, a, a^2, a^3, \ldots, a^{order(a)-1}\}$

$G = \{1, g, g^2, g^3, \ldots\}$

$\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$

$= \{1, g, g^2, \ldots, g^{p-2}\}$ for some $g \in \mathbb{Z}_p^*$

Let $p = 2p' + 1$ be a safe prime

$|\mathbb{Z}_p^*| = 2p'$ $\quad \phi(2p') = p' - 1$

$QR = \{x^2 \mid x \in \mathbb{Z}_p^*\}$

Key Agreement based on Diffie-Hellman

A		B
$Gen(1^\lambda) \to G, g$		
	$\xrightarrow{G, g}$	
$x \xleftarrow{\$} [\lvert G \rvert]$		$y \xleftarrow{\$} [\lvert G \rvert]$
	$\xrightarrow{g^x}$	
	$\xleftarrow{g^y}$	
output $(g^y)^x$		output $(g^x)^y$

Key agreement   Key exchange

from "weak" security ① $\Longrightarrow$ "strong" security ②

- Random Oracle

- Randomness Extractor

- New assumptions

$$\mathrm{Ext}: R \times S \rightarrow \{0,1\}^\lambda$$

$\forall$ distribution D over R satisfying $H_{min}(D) \geq 2\lambda$

$$\mathrm{Ext}(\underbrace{r}_{\text{imperfect randomness}}, \underbrace{s}_{\text{seed perfect random}}) \rightarrow \underbrace{r'}_{\text{perfect randomness}}$$

$$(\mathrm{Ext}(r,s), s) \approx_s (u, s)$$

statistical distance $\leq 2^{-\lambda}$

$r \leftarrow D, \; s \xleftarrow{\$} S$

Computational Diffie-Hellman Assumption

CDH

$Gen(1^\lambda) \to G, g$

$\forall$ p.p.t. A

$x, y \xleftarrow{\$} \{0, 1, \dots, |G|-1\}$

$\Pr[A(G, g, g^x, g^y) = g^{xy}] \le negl(\lambda)$

Decisional Diffie Hellman Assumption

DDH

$Gen(1^\lambda) \to G, g$

$x, y, z \xleftarrow{\$} \{0, 1, \dots, |G|-1\}$

$(G, g, g^x, g^y, g^{xy}) \approx_c (G, g, g^x, g^y, g^z)$

DDH $\Longrightarrow$ CDH

CDH $\not\Longrightarrow$ DDH

E.g. $Gen(1^\lambda) \to (\mathbb{Z}_p^*, g)$

p is a λ-bit safe prime

DDH may hold for

$Gen(1^\lambda) \to (QR_p, g)$ p is a λ-bit safe prime

$QR_p = \{x^2 \mid x \in \mathbb{Z}_p^*\}$

$= \{x \in \mathbb{Z}_p^* \mid x^{\frac{p-1}{2}} = 1\}$

$\forall x \in \mathbb{Z}_p^*$

$x = a^2 \Longleftrightarrow x^{\frac{p-1}{2}} = 1$

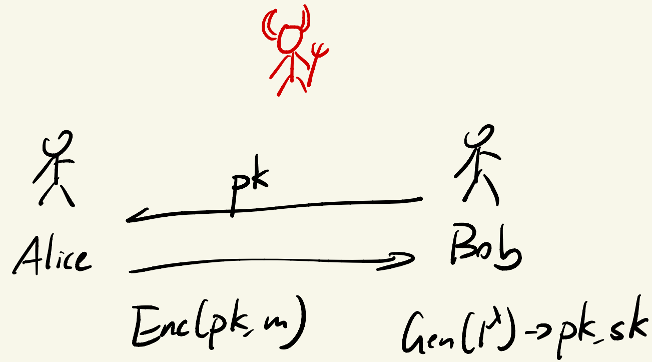
Public-Key Encryption

(Gen, Enc, Dec)

$\mathrm{Gen}(1^\lambda) \to pk, sk$ — pk: public key; sk: secret key

$\mathrm{Enc}(pk, m) \to c$

$\mathrm{Dec}(sk, c) \to m$



Security

$\mathcal{A}$ — Challenger

Challenger: $pk, sk \leftarrow \mathrm{Gen}(1^\lambda)$

Challenger → $\mathcal{A}$: pk

$\mathcal{A}$ → Challenger: m_0, m_1

Challenger: $b \xleftarrow{\$} \{0,1\}$

Challenger → $\mathcal{A}$: $\mathrm{Enc}(pk, m_b)$

$\mathcal{A}$: guess b'

$\mathcal{A}$ wins $b = b'$

El Gamal Encryption

$\mathrm{Gen}(1^\lambda) \to pk = (G, g, g^x) \quad sk = x$

$\mathrm{Enc}(pk, m) \to (g^y, g^{xy} \cdot m)$ $\boxed{\text{assume } m \in G}$

$\mathrm{Dec}(sk, c) = (g^{xy} \cdot m) / (g^y)^x$

Lamport? Key-agreement /exchange based on RO /hash function.

honest parties takes time T,

adversary has running time $\ll T^2$

$$H: [T^2] \to [T^4]$$

Alice samples $a_1 \cdots a_T \leftarrow [T^2]$

computes $H(a_i)$

sends $H(a_i)$

$$\xrightarrow{H(a_1) \cdots H(a_T)}$$

$$\xleftarrow[H(b_1) \cdots H(b_T)]{}$$

Bob sample $b_1 \cdots b_T \in [T^2]$

compute $H(b_i)$

sends $H(b_i)$

with constant probability

$H(a_i) = H(b_j)$

Alice outputs a_i

Bob outputs b_j

Public-key assumption $\Longrightarrow$ CRHF

$$Gen(1^\lambda) \to pp$$

$$CRHF(pp, x) \to H_{pp}(x)$$

(G, g) s.t. Dlog is hard.

$h \leftarrow$ challenge

$$Gen(1^\lambda) \to pp = (g, h)$$

$G = \mathbb{Z}_p^*$

$$CRHF(pp, (x_1, x_2)) = g^{x_1} h^{x_2}$$

$$g^{x_1} h^{x_2} = g^{y_1} h^{y_2}$$

$$g^{x_1 - y_1} = h^{y_2 - x_2}$$

find d s.t. $(y_2 - x_2) d \equiv 1 \mod |G|$

$$g^{(x_1 - y_1)d} = g^{\frac{x_1 - y_1}{y_2 - x_2}} = h$$