CS 127/CSCI E-127: Introduction to Cryptography

Prof. Salil Vadhan Fall 2013

Lecture Notes 14:

Public-Key Encryption

Reading.

- Katz–Lindell 1st edition §9.0-9.3, 10.0-10.2

1 Setting

- Can parties communicate privately *without meeting in advance*? One of the drawbacks of private-key encryption is the exchange of the secret key: the parties have to meet in advance or use some more secure channel.
- Classical view: ability to encrypt ≡ ability to decrypt ≡ possession of key
- Diffie–Hellman '76: *public-key encryption* — separate encryption & decryption keys
  - *public key* = encryption key, anyone can encrypt since the public key is published in some public directory
  - *secret key* = decryption key
  - secret key $\mapsto$ public key should be infeasible
  - We can imagine that there exists a public directory containing everyone's public key: pk_{Alice},pk_{Bob},... To send a message m to Alice, we get pk_{Alice} from the directory and send $\mathsf{Enc}_{pk_{\text{Alice}}}(m)$.
  - **Q:** potential issues with such a public-key infrastructure?
- **Definition 1** *A* public-key encryption scheme *consists of three polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, as follows:*
  - *The* key generation *algorithm* Gen *is a randomized algorithm that takes a security parameter* 1^n *as input returns a pair* (pk, sk)*, where* pk *is the* public key *and* sk *is the* secret key*; we write* $(pk, sk) \xleftarrow{R} \mathsf{Gen}(1^n)$.
  - *The* encryption *algorithm* Enc *is a* stateless *randomized algorithm that takes the public key* pk *and a* plaintext *(aka* message*)* m *and outputs a ciphertext* c*; we write* $c \xleftarrow{R} \mathsf{Enc}_{pk}(m)$.
  - *The* decryption *algorithm* Dec *is a deterministic algorithm that takes the secret key* sk *and a ciphertext* c *and returns a plaintext* $m = \mathsf{Dec}_{sk}(c)$.

  Associated with the scheme is a *plaintext space* $\mathcal{M}_n$ from which m is allowed to be drawn. The message space is a set, often the set of strings of a given length. We require $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$ for all $m \in \mathcal{M}$ and all $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.

2 Security

- All definitions of security in the private-key case extend naturally to public-key case — just provide the adversary with the public key. For example, the eavesdropping adversarial indistinguishability game is as follows:

- All of the relations we have established between notions of security extend to the public-key case, such as the equivalence of indistinguishability and semantic security. Hence public-key encryptions satisfying the above definition are often referred to as *semantically secure* encryption schemes in the literature.

- **Lemma 2** *If public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has indistinguishable encryptions in the sense of Definition 1, then it satisfies indistinguishability under chosen plaintext attack (and hence satisfies multiple-message indistinguishability).*

  Proof sketch:

- **Corollaries:**
  - No deterministic public-key encryption scheme can be secure.
  - No public-key encryption scheme can be perfectly (or even statistically) secure.

3 Trapdoor Permutations

Definition 3 $\mathcal{F} = \{f_i : \mathcal{D}_i \to \mathcal{D}_i\}_{i \in \mathcal{I}}$ *is a* collection of trapdoor permutations *if each* f_i *is a permutation and:*

1. *There is a PPT* $\mathsf{Gen}(1^n)$ *that outputs a pair* (i, t)*, where* $i \in \mathcal{I}$ *is the (public)* key *(or* index*) and* t *is the* trapdoor*.*
2. *Given* i*, one can sample uniformly from* $\mathcal{D}_i$ *in polynomial time.*
3. *Given* i *and* $x \in \mathcal{D}_i$*, one can evaluate* $f_i(x)$ *in polynomial time.*
4. *For every (nonuniform) PPT* A*, there is a negligible function* ε *such that*

$$\Pr\left[A(1^n, I, f_I(X)) \in f_I^{-1}(f_I(X))\right] \leq \varepsilon(n) \qquad \forall n$$

   where the probability is taken over $(I, T) \xleftarrow{R} \mathsf{Gen}(1^n)$*,* $X \xleftarrow{R} \mathcal{D}_I$*, and the coin tosses of* A*.*
5. *Given* t *and* $y \in \mathcal{D}_i$*, one can evaluate* $f_i^{-1}(y)$ *in polynomial time.*

Examples : there are many fewer candidates than OWF

1. RSA: $f_{N,e}(x) = x^e \bmod N$, where $N = pq$, $e \in \mathbb{Z}^*_{\phi(N)}$.

   Trapdoor:

   Inverse map:

   Key generation:

2. Rabin: we have to restrict to the case where Rabin's functions f_N are permutations, i.e. $N = pq$ for $p, q \equiv 3 \bmod 4$ and the domain is QR_N.

   Trapdoor:

   Inverse map: