Lecture 36: Trapdoor One-way Function/Permutation Family

Recall I

Let us recall the RSA assumption and some of its salient properties

- Pick $p, q \xleftarrow{\$} P_n$
- Compute $N = p \cdot q$
- Let $\varphi(N) = \left|\mathbb{Z}_N^*\right| = (p-1)(q-1)$

Claim

For every $e \in \mathbb{Z}^*_{\varphi(N)}$, the function $x^e : \mathbb{Z}^*_N \to \mathbb{Z}^*_N$ is a bijection

Claim (RSA Assumption)

When $e \xleftarrow{\$} \mathbb{Z}^*_{\varphi(N)}$, the function x^e is hard to invert

Note that we are not claiming that x^e is hard to invert for every $e \in \mathbb{Z}^*_{\varphi N}$. There can be choices of e for which it might be easy to

Recall II

invert. But for a randomly chosen $e \xleftarrow{\$} \mathbb{Z}^*_{\varphi(N)}$ the function is hard to invert.

Claim

For each $e \in \mathbb{Z}^*_{\varphi(N)}$ there exists (an associated) d such that $ed = 1 \mod \varphi(N)$. Given d it is easy to invert the function x^e

Recall III

Let us rewrite the previous observation in the following manner

- Let $\mathbb{Z}^*_{\varphi(N)} = \{e_1, e_2, \ldots, e_\alpha\}$
- Consider the family of functions

$$\mathcal{F} = \{f_1 = x^{e_1}, f_2 = x^{e_2}, \ldots, f_\alpha = x^{e_\alpha}\}$$

- Let us write the trapdoor set as

$$T = \{d_1, d_2, \ldots, d_\alpha\}$$

- Each function $f_i = x^{e_i}$ has an associated trapdoor $\text{trap}_i = d_i$ such that, given d_i, it is easy to invert the function $f_i = x^{e_i}$
- When $i \xleftarrow{\$} \{1, 2, \ldots, \alpha\}$, the function f_i is hard to invert
- Given the associated trapdoor d_i, it is easy to invert the function f_i

Definition (Trapdoor OWF)

Let $\mathcal{F}$ represent the following family of functions

$$\mathcal{F} = \{f_1, f_2, \ldots, f_\alpha\},$$

where each function $f_i : \mathcal{D} \to \mathcal{R}$. Let T represent the corresponding set of trapdoors

$$T = \{\text{trap}_1, \text{trap}_2, \ldots, \text{trap}_\alpha\}$$

The family of function $\mathcal{F}$ along with the set of trapdoors T is a trapdoor one-way function family if it satisfies the following conditions.

1. **One-way.** For $i \xleftarrow{\$} \{1, 2, \ldots, \alpha\}$ the function f_i is hard to invert
2. **Trapdoor.** For every $i \in \{1, 2, \ldots, \alpha\}$, given the trapdoor trap_i, it is easy to invert the function f_i

Trapdoor One-way Functions II

- If $\mathcal{D} = \mathcal{R}$ and each f_i is a bijection, then the family $\mathcal{F}$ along with the set of trapdoors T is referred to as a "trapdoor one-way permutation family"
- A particular instance of trapdoor one-way permutation family is provided by the RSA assumption. Note that the "Recall slides" demonstrate that the set of function x^e, where $e \in \mathbb{Z}^*_{\varphi(N)}$ is such a family. The trapdoor for x^{e_i} is d_i such that $e_i d_i = 1 \mod \varphi(N)$.
- Several contexts where RSA is used, we can use any trapdoor one-way permutation family instead. We demonstrate this using RSA key-agreement protocol in the next slide.

Key-agreement using Trapdoor Permutation Family I

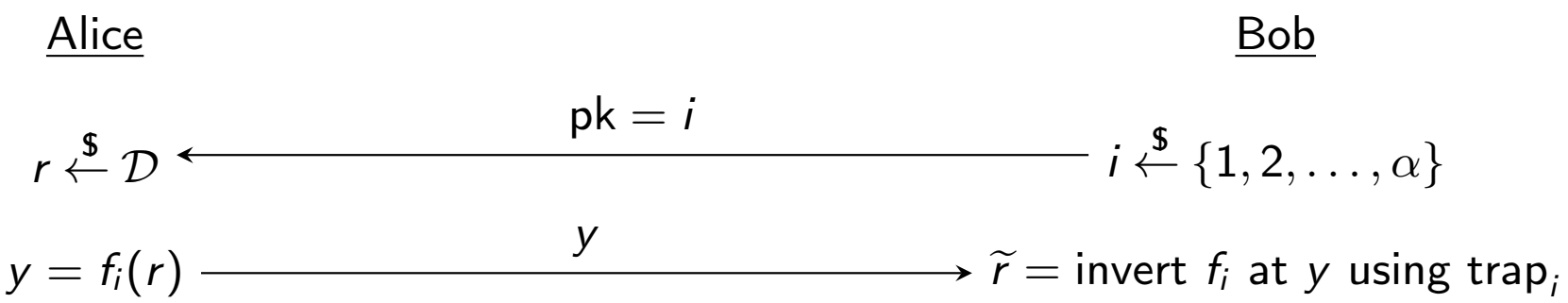
Let us revisit the RSA key-agreement protocol

Alice		Bob
		$p, q \xleftarrow{\$} P_n$
		$N = p \cdot q$
$r \xleftarrow{\$} \mathbb{Z}_N^*$	$\xleftarrow{\text{pk} = (N, n, i)}$	$e_i \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}^* = \{e_1, e_2, \ldots, e_\alpha\}$
$y = r^{e_i}$	$\xrightarrow{y}$	d_i s.t. $e_i d_i = 1 \mod \varphi(N)$
		$\widetilde{r} = y^{d_i}$

Since, each x^e is a permutation we have $r = \widetilde{r}$. The security of this scheme stems from the fact that for $e_i \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}^*$ the image y is hard to invert (for a random $r \xleftarrow{\$} \mathbb{Z}_N^*$)

Key-agreement using Trapdoor Permutation Family II

We can instead write the same scheme using a general trapdoor one-way permutation family. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_\alpha\}$ be a family of functions from the domain $\mathcal{D}$ and range $\mathcal{R}$, and with associated set of trapdoors $\{\text{trap}_1, \text{trap}_2, \ldots, \text{trap}_\alpha\}$.



Key-agreement using Trapdoor Permutation Family III

- At an abstract level, the party that can perform a task and that particular task cannot be performed by an adversary is the holder of the trapdoor. For example, in the previous case, we want Bob to decrypt and an adversary should not be able to decrypt. So, Bob should be the one with the trapdoor.

Let us apply this intuition to solve a new problem

Digital Signatures I

1. In a digital signature scheme the signer publishes a public-key pk
2. Later if the signer wants to endorse a document m, then she signs it with a signature σ
3. Everyone should be able to verify that "the person who published pk" is indeed the one who endorses m
4. An adversary, who gets to see (pk, m, σ) should not be able to forge signatures on new messages. That is, the adversary should not be able to generate (m', σ') that verifies.

Digital Signatures II

- Note that the signer should be able to sign while the adversary should not be able to sign. This means that the signer should have the trapdoor.
- Further, it should be hard for an adversary to sign messages. This implies that "signing a message" should be associated to "inverting the function" (a difficult task if one does not have the trapdoor).

Digital Signatures III

- So, signer should pick an index i
- The signature of m is $\sigma = f_i^{-1}(m)$, which can be easily computed given d_i
- A verifier should be able to test $f_i(\sigma) = m$ (an easy task for everyone). So, the public-key should be $\mathsf{pk} = i$

Digital Signatures IV

Let us write down our new signature scheme

Signer		Verifier
$i \xleftarrow{\$} \{1, \ldots, \alpha\}$	$\xrightarrow{\mathsf{pk} = i}$	
$\sigma = f_i^{-1}(m, \mathsf{trap}_i)$	$\xrightarrow{(m, \sigma)}$	Verify σ is equal to $f_i(m)$

Note that we only need "trapdoor one-way function family" and not "trapdoor one-way permutation family" (because the message is sent in the open, we need not recover it).
We can instantiate this using the RSA trapdoor one-way permutation family and we can get a signature scheme using RSA signature

Digital Signatures V

BUT THIS SCHEME IS INSECURE!!!

Two Attacks.

- Note that an adversary can pick any x and compute $y = f_i(x)$. This implies that $(m' = y, \sigma' = x)$ is a valid forgery (although the adversary does not have control over what message it is signing, this is a valid forgery). This attack can be performed just after seeing the public-key, and the adversary does not need to see any message-signature pairs!
- The scheme using RSA trapdoor one-way permutation family has another attack. Suppose the adversary sees two message-signature pairs (m_1, σ_1) and (m_2, σ_2). Note that $(m' = m_1 \cdot m_2, \sigma' = \sigma_1 \cdot \sigma_2)$ is a valid forgery!

Digital Signatures VII

Students in class proposed the following elegant fix.

- The public-key of the new scheme is $\text{pk} = (i, r)$, where r is a random string of appropriate length
- The signature of the message m is $\sigma = f_i^{-1}(r \| m)$
- The message-signature pair (m, σ) are verified by checking whether $r \| m$ is equal to $f_i(r)$

Consider the following heuristic argument demostrating how this scheme protects against the two attacks mentioned above.

- To perform an attack similar to the first attack discussed above, the adversary will have to pick x such that $f_i(x)$ has r in the prefix, which should be difficult if r is sufficiently long
- To perform an attack similar to the second attack discussed above, the adversary should pick m_1 and m_2 such that $(r \| m_1) \cdot (r \| m_2)$ has r in the prefix, which should again be unlikely