CS 7880 Graduate Cryptography November 11, 2015

Lecture 11: Signatures and Trapdoor Permutations

Lecturer: Daniel Wichs *Scribe: Tanay Mehta*

1 Topics Covered

- Squaring modulo N
- Rabin trapdoor permutation
- Signatures using TDPs in the RO model

2 Rabin Trapdoor Permutation

Last time, we described the RSA cryptosystem. Specifically, we looked at the function

$$f_e(x) = x^e \mod N$$

where $N = p \cdot q$ for primes p, q and $e \in \mathbb{Z}^*_{\varphi(N)}$. In using f_e to encrypt messages, we are making a tautological assumption that f_e is one-way. This relies on the computation hardness of factoring. However, we do not whether the only way to invert f_e is by computing $e^{-1} \mod \varphi(N)$. There might be a different method to break the encryption that does not rely on the discrete log assumption. We will take a look at another function that relies on the weaker assumption that it is difficult to compute square roots modulo N (as opposed to computing e^{th} roots modulo N in the RSA system). In fact, we will show that this assumption is equivalent to hardness of factoring.

As a short detour, we will briefly discuss the new presumed result by Laszlo Babai that the graph isomorphism problem can be solved in quasipolynomial time. One question that arises is why is it believed that graph isomorphism can be solved in polynomial time, but not factoring when both problems are currently thought to be in NP but not NP-Complete? The intuitive answer is that factoring is a distribution of problems with no heuristics, whereas there are heuristics for solving graph isomorphism instances efficiently.

Let us return back to defining a function that is exactly as hard to invert as factoring. The *Rabin trapdoor permutation* is defined to be

$$f(x) = x^2 \mod N$$

There is an immediate problem with this definition — f is not a permutation on $\mathbb{Z}^*_n$. Its image is a subset of $\mathbb{Z}^*_n$. To be able to fix this, let us look more closely at squaring modulo N.

2.1 Squaring Modulo N

Recall that the Chinese remainder theorem says we can study values modulo $N = p \cdot q$ as values modulo p and modulo q. Let us consider the square function modulo p.

$$f: \mathbb{Z}_p^* \to \mathbb{Z}_p^*$$
$$f(x) = x^2 \mod p$$

The image of f is the set of *quadratic residues* modulo p.

$$Im(f) = \boldsymbol{QR}_p = \{y \in \mathbb{Z}_p^* \ : \ \exists x \in \mathbb{Z}_p^* \ y = x^2\}$$

Let us take a closer look at how the elements of $\mathbb{Z}_p^*$ behave under the squaring operation. Since $\mathbb{Z}_p^*$ is cyclic, we have that

$$\begin{array}{c}
\mathbb{Z}_p^* =< g >= \{g^0, g^1, g^2, \ldots, g^{\frac{(p-1)}{2}-1}, g^{\frac{(p-1)}{2}}, \ldots, g^{p-2}\} \\
\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \\
\boldsymbol{QR}_p =< g^2 >= \{g^0, g^2, g^4, \ldots, g^{p-3}, g^0 = 1, \ldots\}
\end{array}$$

More specifically, we see that an element of the form g^i is mapped to g^{2i}. This tells us that the quadratic residues $\boldsymbol{QR}_p$ are the even powers of g, and that

$$|\boldsymbol{QR}_p| = \frac{(p-1)}{2}$$

Observe that since $g^{\frac{(p-1)}{2}}$ maps to $g^{(p-1)} = 1 \mod p$ (by Fermat's little theorem) and $g^0 = 1 \mod p$, we have that $g^{\frac{(p-1)}{2}} = -1 \mod p$. Keeping this in mind, we will introduce new notation to determine whether an element y of $\mathbb{Z}_p$ is a quadratic residue.

Define the *Jacobi symbol* (also called the *Legendre symbol* when p is prime) to be

$$J(y) = y^{\frac{(p-1)}{2}} \mod p$$

Note that if $y = g^x$ then

$$J(y) = g^{x\frac{(p-1)}{2}}$$

Therefore, if x is even modulo p, then $g^{x\frac{(p-1)}{2}} = g^{x'(p-1)} = 1 \mod p$ by Fermat's little theorem. If x is odd, then $g^{x\frac{(p-1)}{2}} = g^{x'(p-1)+\frac{(p-1)}{2}} = -1 \mod p$ by the above observation.

Now we will show that the square function is a permutation when restricted to the domain $\boldsymbol{QR}_p$. We do this by giving a method to invert the square function for $p = 3 \mod 4 = 4i+3$. That is, given $y \in \boldsymbol{QR}_p : \ y = x^2 \mod p$, find x. Consider the following:

$$(y^{i+1})^2 = y^{2i+2} = y^{\frac{(p-1)}{2}+1} = (x^2)^{\frac{(p-1)}{2}+1} = 1 \cdot x^2 = y$$

Therefore, $x = \pm y^{i+1}$.

Note that $(-1) = g^{\frac{(p-1)}{2}} \notin \boldsymbol{QR}_p$ since $p = 3 \mod 4$ implies that $\frac{(p-1)}{2}$ is odd. We can infer from this that $x \in \boldsymbol{QR}_p$ if and only if $-x \notin \boldsymbol{QR}_p$.

We have shown that the square function $f: \boldsymbol{QR}_p \to \boldsymbol{QR}_p$ is a permutation. Note that f is not one-way. As we have just seen, we can invert f easily.

2.2 Rabin Trapdoor Permutation

Let us return to the Rabin trapdoor permutation $f(x) = x^2 \mod N$ where $N = p \cdot q$ for primes $p, q = 3 \mod 4$. Since N is not prime, we will extend the result that the square function is a permutation.

Using the Chinese remainder representation, f maps $x = (x_p, x_q) \mapsto (x_p^2, x_q^2)$. We again define the image of f to be the quadratic residues modulo N.

$$Im(f) = \boldsymbol{QR}_N = \{y \in \mathbb{Z}_N^* \ : \ \exists x \in \mathbb{Z}_N^* \ x^2 = y\}$$

Note that $y \in \boldsymbol{QR}_N \Leftrightarrow y_p \in \boldsymbol{QR}_p, \ y_q \in \boldsymbol{QR}_q$ where (y_p, y_q) is the Chinese remainder theorem representation of y. From this, we see that

$$f^{-1}(y) = (x_p, x_q), (-x_p, x_q), (-x_p, -x_1), (x_p, -x_q)$$

Exactly one of these four inverses is a quadratic residue modulo N because from our previous observation exactly one of $-x_k, x_k$ is a quadratic residue for $k = p, q$. Therefore, we also have that $|\boldsymbol{QR}_N| = \frac{|\mathbb{Z}_N^*|}{4}$.

We have shown that $f: \boldsymbol{QR}_N \to \boldsymbol{QR}_N$ is a permutation. Now we will show that the security of the Rabin cryptosystem is equivalent to hardness of factoring.

Claim 1 *Given x, z such that $x^2 =_N z^2 =_N y$ and $x \neq \pm z$, N can be factored.*

Proof: Recall that $f^{-1}(y)$ takes on one of four values shown above. As a result, the sum of two distinct preimages x and z yields

$$x + z = (0, 2x_q) \text{ or } (2x_p, 0)$$

Without loss of generality, let $x + z = (2x_p, 0)$. Then, we have that

$$x + z = 0 \mod q$$

$$x + z / neq 0 \mod p$$

Thus, we can extract the value of q by taking $gcd(x + z, N) = q$. $\square$

Theorem 1 *If factoring is hard, inverting $f(x)$ is hard.*

Proof: Assume $\mathcal{A}$ inverts $f(x)$ in poly-time. Then, we have that

$$\Pr[\mathcal{A}(N,y) = x,\ x^2 = y \mod N\ :\ p,q \leftarrow \mathsf{Primes}, N = pq, y \leftarrow \boldsymbol{QR}_N] \geq \varepsilon(n) = \frac{1}{\mathsf{poly}}$$

Consider the following algorithm.

$$\mathsf{Factor}(N) : \begin{cases} \text{Choose a random } x \leftarrow \mathbb{Z}_N^* \\ y = x^2 \\ z = A(N,y) \\ \text{If } z \neq \pm x \text{ then use claim and factor } N \end{cases}$$

The probability that the z returned by $\mathcal{A}$ satisfies the conditions of the claim is $\frac{1}{2}$. Therefore,

$$\Pr[\mathsf{Factor}(N) \text{ succeeds}] \geq \frac{\varepsilon(n)}{2}$$

□

3 Signatures

Let us look at a new application of public-key cryptosystems, signatures. Consider the high-level view of the protocol between Alice and Bob with an eavesdropper Eve.

$$\begin{array}{rcl} \underline{\textit{Alice}} & & \underline{\textit{Bob}} \\ (vk, sk) & & vk \\ \sigma \leftarrow \mathsf{Sign}(sk, m) & & \\ (m, \sigma) & \rightarrow \underline{\textit{Eve}} \rightarrow & (m', \sigma') \\ & & \mathsf{Verify}(vk, m, \sigma) = 0, 1 \end{array}$$

More formally, a signature scheme consists of the following:

- $(vk, sk) \leftarrow \text{KeyGen}(1^n)$: A key generation function generates the verification key vk and the secret key sk.
- $\sigma \leftarrow \text{Sign}(sk, m)$: A signing function that takes a message m and the secret key sk to produce a signature σ.
- $b \leftarrow \text{Verify}(vk, m, \sigma)$: A verification function that takes the verification key vk, message m, and signature σ and returns a bit b.

Correctness is established by ensuring that for all $(vk, sk) \in \mathsf{KeyGen}(1^n)$ and for all $m \in \mathcal{M}$

$$\Pr[\mathsf{Verify}(vk, m, \sigma) = 1\ :\ \sigma \leftarrow \mathsf{Sign}(sk, m)] = 1$$

Security is established by ensuring a probability bound on the following game with an adversary $\mathcal{A}$.

$\mathsf{SigGame}_{\mathcal{A}}(n)$:

$$\begin{array}{rcl}
\underline{\mathsf{Signer}} & & \underline{\mathcal{A}} \\
(vk, sk) \leftarrow \mathsf{KeyGen}(1^n) & & \\
vk & \rightarrow & vk \\
m_1 & \leftarrow & m_1 \\
\mathsf{Sign}(sk, m_1) = \sigma_1 & \rightarrow & \sigma_1 \\
\vdots & \vdots & \vdots \\
m_k & \leftarrow & m_k \\
\mathsf{Sign}(sk, m_k) = \sigma_k & \rightarrow & \sigma_k \\
(m^*, \sigma^*) & \leftarrow & (m^*, \sigma^*)
\end{array}$$

Output 1 if $m^* \notin \{m_i\}_{1 \le i \le k}$ and $\mathsf{Verify}(vk, m^*, \sigma^*) = 1$.

For the above game, we define security to be that for all PPT $\mathcal{A}$

$$\Pr[\mathsf{SigGame}_{\mathcal{A}}(n) = 1] = \mathsf{negl}(n)$$

This type of security is called *Chosen-Message Attack Security.*

Discussion. Conceptually, signatures look at public-key cryptography (for assumptions). However, we do know how to build them from OWFs so they are like symmetric-key cryptography in that respect. However, while symmetric-key cryptography belongs in the minicrypt world (from Impagliazzo's worlds), signatures belong more in the public-key cryptography world, crytomania.

3.1 Construction using TDPs in the RO Model

We will work towards a construction for signatures using trapdoor permutations in the random oracle model. Let's start with a not quite correct construction.

Let $f_{pk} : D_{pk} \rightarrow D_{pk}$ be a trapdoor permutation with public-key pk. Let Inv_{sk} be its inverse with secret-key sk. Define for the signature scheme:

$$\begin{aligned}
&vk = pk \\
&sk = sk \\
&\mathsf{Sign}(sk, m) = \mathsf{Inv}_{sk}(m) \\
&\mathsf{Verify}(pk, m, \sigma) : \text{Check } f_{pk}(\sigma) = m
\end{aligned}$$

This scheme can be broken by choosing a signature σ first, and then computing and sending $f_{pk}(\sigma) = m$. We can get around this vulnerability by disallowing the adversary $\mathcal{A}$ to choose

m^* but rather sample m^* randomly.

However, the issues with this scheme don't end there. There exists a method where an adversary can sign messages related to ones that are seen. Consider the case where we use the RSA trapdoor permutation.

$$f_{pk}(\sigma) = m \Rightarrow f_{pk}(\sigma \cdot m) = m(x')^e$$

Therefore, given a message m_1 and its signature, an adversary can sign the product of messages $m_1 \cdot m_2$.

Within the random oracle model (in practice via a hash function), there is a simple fix to these vulnerabilities.

$$\begin{aligned} &\mathsf{Sign}(sk, m) = \mathsf{Inv}_{sk}(RO(m)) \\ &\mathsf{Verify}(pk, m, \sigma) : \text{Check } f_{pk}(\sigma) = RO(m) \end{aligned}$$

The first vulnerability is fixed since choosing a signature σ and then computing $f_{pk}(\sigma)$ gives the hash of some message, not the message itself. Therefore, $\mathcal{A}$ cannot produce a message to go along with the signature σ. For the second vulnerability, the above implication no longer holds.