Modern Cryptography January 16, 2019

Solutions to Homework 11

Lecturer: Daniel Slamanig, TA: Karen Klein

1. Key Exchange

   - **[10.4 in book, 2nd edition]** Consider the following key-exchange protocol:
     - Alice chooses uniform $k, r \in \{0,1\}^n$, and sends $s := k \oplus r$ to Bob.
     - Bob chooses uniform $t \in \{0,1\}^n$, and sends $u := s \oplus t$ to Alice.
     - Alice computes $w := u \oplus r$ and sends w to Bob.
     - Alice outputs k and Bob outputs $w \oplus t$.

     Show that Alice and Bob output the same key. Analyze the security of the scheme (i.e., either prove its security or show a concrete attack).

     Solution: First, we show that Alice and Bob output the same key k:

     $$w \oplus t = u \oplus r \oplus t = s \oplus t \oplus r \oplus t = s \oplus r = k \oplus r \oplus r = k.$$

     In the key exchange security game $\widehat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ an adversary $\mathcal{A}$ gets to see the transcript $trans = (s, u, w)$ and a key k^* where k^* either is the real key k (if $b = 0$) or a uniformly random string in $\{0,1\}^n$ (if $b = 1$). In the end of the game, $\mathcal{A}$ outputs a bit b^* and he wins the game if $b^* = b$. The key exchange protocol Π is called secure in the presence of an eavesdropper, if for every PPT adversary $\mathcal{A}$ there exists a negligible function negl such that

     $$\Pr[b^* = b] \leq \frac{1}{2} + \mathsf{negl}(n).$$

     The above protocol is clearly not secure. To see this, note that

     $$s \oplus u \oplus w = (k \oplus r) \oplus (k \oplus r \oplus t) \oplus (k \oplus r \oplus t \oplus r) = k.$$

     Thus, we construct an adversary $\mathcal{A}$ as follows: First, $\mathcal{A}$ computes $k' = s \oplus u \oplus w$. Then he outputs $b^* = 0$ if $k^* = k'$, and $b^* = 1$ else. Obviously, $\mathcal{A}$ wins the game except for the case where $b = 1$ and the uniformly random key k^* happens to coincide with the real key k. Since $\Pr[k^* = k | b = 1] = \frac{1}{2^n}$, we can compute $\mathcal{A}$'s success probability as $\Pr[b^* = b] = 1 - \frac{1}{2^{n+1}}$ which is clearly larger than $\frac{1}{2} + \mathsf{negl}(n)$ for any negligible function $\mathsf{negl}(n)$. □

2. Textbook RSA encryption

   - Prove the correctness of the textbook RSA encryption algorithm as introduced in the lecture, i.e., show that for all $n \in \mathbb{N}$, $((d, N), (e, N)) \leftarrow \mathsf{KeyGen}(1^n)$ any $m \in \mathbb{Z}_N$ it holds that $(m^e)^d \equiv m \pmod{N}$.

Solution: By the chinese remainder theorem, we know that $f : \mathbb{Z}_N \to \mathbb{Z}_p \times \mathbb{Z}_q$, $f(x) = ([x \bmod p], [x \bmod q])$ is a group isomorphism. It is easy to show, that f also preserves the multiplicative structure[1]: Let $x, y \in \mathbb{Z}_N$, then

$$f(xy) = ([xy \bmod p], [xy \bmod q]) = ([[x \bmod p]\cdot[y \bmod p] \bmod p], [[x \bmod q]\cdot[y \bmod q] \bmod q])$$

$$= ([x \bmod p], [x \bmod q]) \cdot ([y \bmod p], [y \bmod q]) = f(x) \cdot f(y).$$

For $x_p \in \mathbb{Z}_p^*$, $i \in \mathbb{Z}$, it holds $x_p^i = x_p^{i \bmod (p-1)} \bmod p$ since $|\mathbb{Z}_p^*| = p-1$. On the other hand, also $0^i = 0 = 0^{i \bmod (p-1)} \bmod p$ for all $i \in \mathbb{Z} \setminus (p-1)\mathbb{Z}$ [2] and in particular for all $i \in \mathbb{Z}$ such that $\gcd(i, \varphi(N)) = 1$, so it holds $x_p^i = x_p^{i \bmod (p-1)} \bmod p$ for all $x_p \in \mathbb{Z}_p$, $i \in \mathbb{Z}$ such that $\gcd(i, \varphi(N)) = 1$. For $k \in \mathbb{N}$ it follows $x^i = x_p^{i \bmod (p-1)} = x_p^{i \bmod k(p-1)} \bmod p$ for all $x_p \in \mathbb{Z}_p$. Similar relations hold in $\mathbb{Z}_q$. For $x \in \mathbb{Z}_N$, $x_p = [x \bmod p], x_q = [x \bmod q]$, and $i \in \mathbb{Z}$ such that $\gcd(i, \varphi(N)) = 1$ it follows:

$$x^i = f^{-1}(f(x^i)) = f^{-1}(f(x)^i) = f^{-1}([x_p^i \bmod p], [x_q^i \bmod q]) =$$

$$f^{-1}([x_p^{i \bmod (p-1)(q-1)} \bmod p], [x_q^{i \bmod (p-1)(q-1)} \bmod q])$$
$$= f^{-1}(f(x)^{i \bmod \varphi(N)}) = x^{i \bmod \varphi(N)} \bmod N.$$

We now prove correctness of textbook RSA. The algorithm KeyGen picks a uniformly random element e in $\mathbb{Z}_{\varphi(N)}^*$, computes its inverse $d := e^{-1} \bmod \varphi(N)$ and outputs $(sk, pk) = ((d, N), (e, N))$. In particular, we have $ed = de = 1 \bmod \varphi(N)$. This implies for any message $m \in \mathbb{Z}_N$:

$$\mathsf{Dec}(\mathsf{Enc}(m, pk), sk) = \mathsf{Dec}([m^e \bmod N]), sk) = [[m^e \bmod N]^d \bmod N] = [(m^e)^d \bmod N]$$

$$= [m^{ed} \bmod N] = [m^{[ed \bmod \varphi(N)]} \bmod N] = [m^1 \bmod N] = m.$$

□

- Show that factoring an RSA integer $N = pq$ is equivalent to computing the order $\varphi(N)$ of the group $\mathbb{Z}_N^*$. Use this result to show that an efficient algotithm for factoring yields an efficient algorithm for solving RSA.

Solution: For an RSA modulus $N = pq$ it holds $\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$. Thus, any PPT algorithm $\mathcal{A}$ for factoring an RSA integer N trivially leads to a PPT algorithm $\mathcal{A}'$ for computing $\varphi(N)$ and $\mathcal{A}'$ has the same success probability as $\mathcal{A}$. On the other hand, let $\mathcal{A}$ be a PPT algorithm for computing $\varphi(N)$ for an RSA integer N. Then we define an algorithm $\mathcal{A}'$ for factoring as follows: Given an RSA modulus N, first $\mathcal{A}'$ runs $\mathcal{A}$ on N to obtain an integer ν. An integer π is a nontrivial factor of N, i.e., $\pi = p$ or $\pi = q$, if and only if π is a solution to $\varphi(N) = N - \pi - N/\pi + 1$. Multiplying this equation by π leads to the quadratic equation $\pi^2 - (N - \varphi(N) + 1)\pi + N = 0$ with the two solutions $\pi = p$ and $\pi = q$. Thus, $\mathcal{A}'$ proceeds by solving the quadratic equation

[1] Note, for the restriction $f|_{\mathbb{Z}_N^*}$ this is already known by the chinese remainder theorem.

[2] Note, for $i \in (p-1)\mathbb{Z}$ it holds $[i \bmod (p-1)] = 0$ and hence $0^i = 0 \neq 1 = 0^0 = 0^{i \bmod (p-1)}$. On the other hand, for all $i \in \mathbb{Z} \setminus (p-1)\mathbb{Z}$ we have $[i \bmod (p-1)] > 0$ and the equality is satisfied.

$\pi^2 - (N - \nu + 1)\pi + N = 0$ in the variable π and outputs its two solutions. $\mathcal{A}'$ succeeds in factoring N whenever $\mathcal{A}$ succeeds in computing $\varphi(N)$. This proves equivalence of factoring $N = pq$ and computing $\varphi(N)$.

We now use this result to show that any efficient algorithm for factoring yields an efficient algorithm for solving RSA. Let $\mathcal{A}$ be an efficient algorithm for factoring and (N, e, y) an RSA instance. We construct an efficient algorithm for RSA as follows: First, $\mathcal{A}'$ queries $\mathcal{A}$ on N and receives two integers π, π'. If $\pi\pi' = N$, then $\mathcal{A}$ computes $\varphi(N)$ and computes $d := e^{-1} \bmod \varphi(N)$. It outputs $x := y^d \bmod N$. Otherwise, $\mathcal{A}'$ outputs a uniform $x \in \mathbb{Z}_N^*$. The success probability of $\mathcal{A}'$ can thus be lowerbounded by the success probability of $\mathcal{A}$:

$$\Pr[\mathsf{RSA} - \mathsf{Inv}_{\mathcal{A}',\mathsf{GenRSA}}(n) = 1] \geq \Pr[\mathsf{Factoring}_{\mathcal{A},\mathsf{GenModulus}}(n)].$$

$\square$

3. IND-CPA secure encryption in the ROM

- **[11.19 in book, 2nd edition]** Say three users have RSA public keys $(3, N_1)$, $(3, N_2)$, and $(3, N_3)$ (i.e., they all use $e = 3$), with $N_1 < N_2 < N_3$. Consider the following method for sending the same message $m \in \{0,1\}^\ell$ to each of these parties: choose a uniform $r \leftarrow \mathbb{Z}_{N_1}^*$, and send to everyone the same ciphertext

  $$(c_1, c_2, c_3, c_4) := (r^3 \bmod N_1, r^3 \bmod N_2, r^3 \bmod N_3, H(r) \oplus m)$$

  where $H : \mathbb{Z}_{N_1}^* \to \{0,1\}^\ell$. Assume $\ell \gg n$.

  - Show that this is not IND-CPA-secure, and an adversary can recover m from the ciphertext even when H is modeled as a random oracle (Hint: Chinese remainder theorem).

    Solution: If N_1, N_2, N_3 are not pairwise coprime, then there are $i, j \in \{1, 2, 3\}$, $i \neq j$ such that $\gcd(N_i, N_j)$ is a nontrivial factor of N_i, hence the adversary can factor N_i and solve RSA as shown in exercise 2. Thus, in this case it is easy to recover r from c_i. The adversary then queries the random oracle on input r and computes $m = c_4 \oplus H(r)$. Now assume N_1, N_2, N_3 are pairwise coprime. Then by the Chinese remainder theorem it holds $Z_{N_1 N_2 N_3}^* \simeq Z_{N_1}^* \times Z_{N_2}^* \times Z_{N_3}^*$ where the isomorphism is given as $f(x) = ([x \bmod N_1], [x \bmod N_2], [x \bmod N_3])$ and can be efficiently inverted. Thus, an adversary can compute $[r^3 \bmod N_1 N_2 N_3] = f^{-1}(c_1, c_2, c_3)$. Since $r \in \mathbb{Z}_{N_1}^*$, we have $0 < r < N_1$, which implies $0 < r^3 < N_1^3 < N_1 N_2 N_3$ and, hence, $[r^3 \bmod N_1 N_2 N_3] = r^3$ is a cube in $\mathbb{Z}$. This implies, that an adversary can recover r by simply computing the cube root of $[r^3 \bmod N_1 N_2 N_3]$ in $\mathbb{Z}$, which can be done efficiently. $\square$

  - Show a simple way to fix this and get a IND-CPA-secure method that transmits a ciphertext of length $3\ell + \mathcal{O}(n)$ (you do not need to provide a formal proof of IND-CPA security).

    Solution: An easy way to fix this is to choose three independent values $r_1, r_2, r_3 \leftarrow \mathbb{Z}_{N_1}^*$ and send the ciphertext

    $$(c_1, c_2, c_3, c_4, c_5, c_6) := \begin{pmatrix} [r_1^3 \bmod N_1], & [r_2^3 \bmod N_2], & [r_3^3 \bmod N_3], \\ H(r_1) \oplus m, & H(r_2) \oplus m, & H(r_3) \oplus m \end{pmatrix}.$$

$\square$

- Show a better approach that is still IND-CPA-secure but with a ciphertext of length $\ell + \mathcal{O}(n)$ (you do not need to provide a formal proof of IND-CPA security).

  Solution: An easy approach would be to simply use a larger exponent e, e.g., a uniformly random e. Although there is no explicit attack known, there doesn't seem to be a simple proof from RSA either. Thus, we will follow a different approach based on hybrid encryption: Let $(\mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure private-key encryption scheme and $H : \mathbb{Z}_{N_1}^* \to \{0,1\}^n$ a random oracle. To send the message $m \in \{0,1\}^\ell$, choose four independent values $r_1, r_2, r_3 \leftarrow \mathbb{Z}_{N_1}^*$, and $k \leftarrow \{0,1\}^n$, and send the ciphertext

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7) := \begin{pmatrix} [r_1^3 \bmod N_1], & [r_2^3 \bmod N_2], & [r_3^3 \bmod N_3], & \\ H(r_1) \oplus k, & H(r_2) \oplus k, & H(r_3) \oplus k, & \mathsf{Enc}_k(m) \end{pmatrix}.$$

$\square$