6.875: Cryptography and Cryptanalysis **October 6, 2021**

Problem Set 3

Instructor: Vinod Vaikuntanathan

TAs: Lali Devadas, Aparna Gupte, Sacha Servan-Schreiber

Instructions.

- **When:** This problem set is due on **October 27, 2021** before **11pm ET**.
- **How:** You should use LaTeX to type up your solutions (you can use our LaTeX template from the course webpage). Solutions should be uploaded on Gradescope as a single pdf file.
- **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions *entirely on your own* and *acknowledge your collaborators.*
- **Reference your sources:** If you use material from outside the lectures, you must reference your sources (papers, websites, wikipedia, ...).
- **When in doubt, ask questions:** Use Piazza or the TA office hours for questions about the problem set. See the course webpage for the timings.

Problem 1. **Find the one-way function!** Almost all cryptographic primitives imply the existence of a one-way function. In this problem, you will prove that both public key and secret key encryption imply the existence of a one-way function.

> **For each of the following cryptographic primitives: assuming you have an arbitrary scheme $\mathcal{E}$ which satisfies perfect correctness and the security definition, construct a one-way function based on $\mathcal{E}$, and prove its one-wayness.**

(a) IND-CPA-secure *public key* encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Solution

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a correct IND-secure PKE scheme. Define $f : \{0,1\}^\lambda \rightarrow \{0,1\}^{|pk|}$ by $f(r_k) = pk_r$ where $(pk_r, sk_r) \leftarrow \mathsf{Gen}(1^\lambda; r_k)$. Suppose towards contradiction that f is not a one-way function. Then there exists some PPT $\mathcal{B}$ which provides a preimage of $f(r_k)$ for uniformly random r_k with non-negligible probability. Define $\mathcal{A}$ which plays the IND-security game against Π using $\mathcal{B}$ as follows:

```
Algorithm $\mathcal{A}_1(1^\lambda, pk)$
$m_0, m_1 \overset{R}{\leftarrow} \{0,1\}^\lambda$
return $m_0, m_1, \mathsf{state} = \{m_0, m_1\}$
```

```
Algorithm $\mathcal{A}_2(1^\lambda, pk, c, \mathsf{state})$
parse $\mathsf{state} = \{m_0, m_1\}$
$r'_k \leftarrow \mathcal{B}(1^\lambda, pk)$
$(pk', sk') \leftarrow \mathsf{Gen}(1^\lambda; r'_k)$
if $pk' \neq pk$ : return $\tilde{b} \overset{R}{\leftarrow} \{0,1\}$
if $\mathsf{Dec}(sk', c) = m_0$ : return 0
else : return 1
```

Analysis. pk was produced by running Gen (with uniformly random r), so it is a valid input to $\mathcal{B}$ for the one-way function game. We have that $pk' = pk \implies c \in \{\mathsf{Enc}(pk', m_b; r^*\}_{r^* \in \{0,1\}^\lambda} \implies \mathsf{Dec}(sk', c) = m_b$ by correctness of decryption, so $\mathcal{A}$ wins the IND-security game whenever $\mathcal{B}$ successfully provides a preimage of $f(r_k)$, and guesses randomly otherwise.
We assumed that $\mathcal{B}$ successfully provides a preimage of $f(r_k)$ for uniformly random r_k with non-negligible probability, so $\mathcal{A}$ wins the IND-security game against Π with non-negligible advantage (assuming $m_0 \neq m_1$, which happens with negligible probability), but this is a contradiction, since we know Π is IND-secure. Thus f is a one-way function.

(b) IND-CPA-secure *secret key* encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Solution

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a correct IND-CPA-secure secret key encryption scheme with key length k, message length λ, and ciphertext length $|c|$. Pick ℓ large enough that $\ell\lambda \geq 2k$ (we will see why later). $\ell = \mathsf{poly}(k)$, so by a hybrid argument like the ones we have seen in class and on HW 1, we can use challenge message sets $\vec{m}_0, \vec{m}_1$ of size ℓ in the IND-CPA security game.

Define $f : \{0,1\}^\lambda \times (\{0,1\}^\lambda)^\ell \times (\{0,1\}^\lambda)^\ell \to (\{0,1\}^{|c|})^\ell$ by $f(r_k, \vec{m}, \vec{r}) = (\vec{m}, \vec{c})$ where $sk \leftarrow \mathsf{Gen}(1^\lambda; r_k)$ and $c_i \leftarrow \mathsf{Enc}(sk, m_i; r_i)$. Suppose towards contradiction that f is not a one-way function. Then there exists some PPT $\mathcal{B}$ which provides a preimage of $f(r_k, \vec{m}, \vec{r})$ for uniformly random $r_k, \vec{m}, \vec{r}$ with non-negligible probability $\delta(\lambda)$. Define $(\mathcal{A}_1, \mathcal{A}_2)$ which plays the IND-security game against Π using $\mathcal{B}$ as follows:

```
Algorithm $\mathcal{A}_1(1^\lambda, pk)$
$\vec{m}_0, \vec{m}_1 \xleftarrow{R} (\{0,1\}^\lambda)^\ell$
return $\vec{m}_0, \vec{m}_1, \mathsf{state} = \{\vec{m}_0, \vec{m}_1\}$
```

```
Algorithm $\mathcal{A}_2(1^\lambda, pk, c, \mathsf{state})$
parse $\mathsf{state} = \{\vec{m}_0, \vec{m}_1\}$
for $b \in \{0,1\}$ :
    $(r_k^b, \vec{m}^b, \vec{r}^{\,b}) \leftarrow \mathcal{B}(1^\lambda, (\vec{m}_b, \vec{c}))$
    if $f(r_k^b, \vec{m}^b, \vec{r}^{\,b}) = (\vec{m}_b, \vec{c})$ : return $b$
return $\tilde{b} \xleftarrow{R} \{0,1\}$
```

Analysis. $\mathcal{A}_2$ succeeds in distinguishing when $\mathcal{B}$ successfully inverts $(\vec{m}_b, \vec{c})$ and fails to invert $(\vec{m}_{1-b}, \vec{c})$. We already know that $(\vec{m}_b, \vec{c})$ is a valid output of f on uniformly random input, and we will set ℓ (the challenge message set size) so that $(\vec{m}_{1-b}, \vec{c})$ is a valid output of f only with negligible probability.

Formally, let us fix $\vec{c}$ and then consider the set of "bad" messages $\vec{m}$ such that $(\vec{m}, \vec{c})$ is a valid output of f, i.e. $M_{\vec{c}} = \{\vec{m} \in (\{0,1\}^\lambda)^\ell \mid \exists\, sk \text{ s.t. } \mathsf{Dec}(sk, \vec{c}) = \vec{m}\}$. Note that $\exists\, x$ s.t. $f(x) = (\vec{m}, \vec{c}) \iff \vec{m} \in M_{\vec{c}}$ and that $|M_{\vec{c}}| \leq \#$ of possible keys $\leq 2^k$. Then

$$\Pr_{\vec{m}_{1-b} \xleftarrow{R} (\{0,1\}^\lambda)^\ell}[\vec{m}_{1-b} \in M_{\vec{c}}] \leq \frac{2^k}{2^{\ell\lambda}} \leq \frac{2^{\ell\lambda/2}}{2^{\ell\lambda}} = \frac{1}{2^{\ell\lambda/2}} = \mathsf{negl}(\lambda)$$

so $(\vec{m}_{1-b}, \vec{c})$ is a valid output of f only with negligible probability, as desired, so $\mathcal{B}$ can only successfully invert $(\vec{m}_{1-b}, \vec{c})$ with negligible probability. On the other hand, $\vec{m}_b \xleftarrow{R} M_{\vec{c}} \implies \mathcal{B}(\vec{m}_b, \vec{c})$ returns a correct preimage with probability $\delta(\lambda)$. Then

$$\begin{aligned}
&\Pr[(\mathcal{A}_1, \mathcal{A}_2) \text{ wins IND-CPA security game}] \\
&\geq \Pr\Big[\Big(\vec{m}_{1-b} \notin M_{\vec{c}} \;\vee\; \big(\vec{m}_{1-b} \in M_{\vec{c}} \;\wedge\; \mathcal{B} \text{ fails on } (\vec{m}_{1-b}, \vec{c})\big)\Big) \\
&\qquad\qquad \wedge\; \Big(\mathcal{B} \text{ succeeds on } (\vec{m}_b, \vec{c}) \;\vee\; \big(\mathcal{B} \text{ fails on } (\vec{m}_b, \vec{c}) \;\wedge\; \text{coin flip is correct}\big)\Big)\Big] \\
&= \Big[(1 - \mathsf{negl}(\lambda)) + \mathsf{negl}(\lambda)(1 - \delta(\lambda)\Big]\Big[\delta(\lambda) + (1 - \delta(\lambda))\frac{1}{2}\Big] \\
&= [1 - \delta(\lambda) \cdot \mathsf{negl}(\lambda)]\left[\frac{1}{2} + \frac{1}{2}\delta(\lambda)\right] = \frac{1}{2} + \frac{1}{2}\delta(\lambda) - \frac{1}{2}\delta(\lambda) \cdot \mathsf{negl}(\lambda) - \frac{1}{2}\delta(\lambda)^2 \cdot \mathsf{negl}(\lambda) \\
&\geq \frac{1}{2} + \frac{1}{2}\delta(\lambda) - \delta(\lambda) \cdot \mathsf{negl}(\lambda) = \frac{1}{2} + \delta(\lambda) \cdot \left(\frac{1}{2} - \mathsf{negl}(\lambda)\right) \\
&= \frac{1}{2} + \delta(\lambda) \cdot \mathsf{nonnegl}(\lambda) \geq \frac{1}{2} + \mathsf{nonnegl}(\lambda)
\end{aligned}$$

but this is a contradiction, since we know Π is IND-secure. Thus f is a OWF.

Problem 2. Validating RSA ciphertexts

Alice is a stock broker communicating with the New York Stock Exchange (NYSE). To prevent a competing trader Eve from learning whether she is issuing BUY or SELL orders to NYSE, all communication to NYSE is encrypted using the RSA scheme described in Figure 1, where NYSE holds the secret key. The messages BUY ($m = 0$) and SELL ($m = 1$) are encoded as a bit which only NYSE can recover using the secret key.

The encryption scheme described below relies on the RSA assumption and uses the fact that the least significant bit (LSB) of an RSA encryption is a hardcore bit.

```
Gen(1^λ)
sample random distinct
      $\lambda$-bit primes $p$ and $q$
$N := pq$
$e \xleftarrow{R} \mathbb{Z}_N^*,\ d := e^{-1} \pmod{\phi(N)}$
$sk := d,\ pk := (N, e)$
return $(sk, pk)$
```

```
Enc($pk, m \in \{0,1\}$)
parse $pk = (N, e)$
$r \xleftarrow{R} \{1, 2, \ldots, 2^{\lambda-1} - 1\}$
$c := (2r + m)^e \pmod{N}$
return $c$
```

```
Dec($pk, sk, c$)
parse $pk = (N, e)$ and $sk = d$
$\tilde{c} := c^d \pmod{N}$
return LSB($\tilde{c}$)
```

Figure 1: The RSA scheme

Corrupted ciphertexts. However, because Alice is forced to use Xfinity (which has notoriously faulty connections), ciphertexts aren't always reliably transmitted to NYSE: some bits of the ciphertexts are flipped with high probability. However, such corruption is *undetectable* by NYSE, who will happily decrypt an (invalid) message from the corrupted ciphertext, e.g., potentially recovering SELL when Alice sent BUY. To prevent the NYSE from recovering an incorrect message when a ciphertext is corrupted, Alice proposes a variant of the RSA scheme with an integrated "message validity check", described in Figure 3. If any bits of the ciphertext c are corrupted (i.e., flipped) before being received by the NYSE, then the goal is to ensure that Dec$(pk, sk, c) = \perp$ (the decryption fails) with overwhelming probability, and NYSE responds by asking the sender to re-transmit her message. Alice's scheme ensures that all valid encryptions have the same format (visualized in Figure 2).

r	0000 ... 0	m
λ-2 bits	λ-1 bits	1 bit

Figure 2: Bit-wise representation of a message encrypted using Alice's variant of the RSA scheme. The randomness is added to make each ciphertext look random while the zeroes are added to detect corruption with high probability; the least significant bit m encodes the BUY or SELL action.

Suppose Eve intercepts an encryption of m under the RSA scheme of Figure 3 sent by Alice before it is corrupted (perhaps she has access to Alice's WiFi router). Provide an attack showing that Eve can recover the message bit

```
Gen(1^λ)
sample random distinct
    $\lambda$-bit primes $p$ and $q$
$N := pq$
$e \xleftarrow{R} \mathbb{Z}_N^*,\ d := e^{-1} \pmod{\phi(N)}$
$sk := d,\ pk := (N, e)$
return $(sk, pk)$
```

```
Enc(pk, m ∈ {0,1})
parse $pk = (N, e)$
$r \xleftarrow{R} \{1, 2, \ldots, 2^{\lambda-2} - 1\}$
$c := (2^\lambda r + m)^e \pmod{N}$
return $c$
```

```
Dec(pk, sk, c)
parse $pk = (N, e)$ and $sk = d$
$\tilde{c} := c^d \pmod{N}$
// change to bit representation
$t := \tilde{c}_1 \ldots \tilde{c}_\ell$ // (LSB is $\tilde{c}_\ell$)
if $\ell < \lambda + 1$ : return $\perp$
if $t_{[\ell-\lambda+1:\ell-1]} \neq 0^{\lambda-1}$ : return $\perp$
else : return $\mathsf{LSB}(\tilde{c})$
```

Figure 3: Alice's RSA scheme with integrated message validity checking

m (encoding BUY or SELL) with $1 - \mathsf{negl}(\lambda)$ probability by interacting with the NYSE. *You may assume that Eve has a perfect internet connection with the NYSE (ciphertexts are never corrupted when sent to NYSE by Eve).*

Solution

Let $c = \mathsf{Enc}(pk, m)$ be a ciphertext issued by Alice and incercepted by Eve. Consider the following attack performed by Eve.

```
Eve(c)
$z \leftarrow c \cdot 2^e \pmod{N}$
send $z$ to NYSE
if NYSE asks for re-transmission :
    return 1
else : return 0
```

Analysis: Consider what happens when the NYSE decrypts z.
We have that for some $r \in \{1, \ldots, 2^{\lambda-2} - 1\}$,

$$\tilde{z} = z^d = \left(\left(2^\lambda r + m\right)^e \cdot 2^e\right)^d = \left(2^\lambda r + m\right)^{ed} \cdot 2^{ed} = 2^{\lambda+1} r + 2m \pmod{N}.$$

If we let t be the bit representation of $c^d = \left(2^\lambda r + m\right)^{ed} = 2^\lambda r + m$, then the bit representation y of $\tilde{z}$ is just t shifted one bit to the left. Then we have that

$$y_{[\ell-\lambda+1:\ell-1]} = t_{[\ell-\lambda+2:\ell]} = 0^{\lambda-2} \mid\mid m,$$

so the NYSE will ask for re-transmission upon receiving z if and only if $m = 1$. As such, Eve learns the message m with probability 1.

Remark 1. *We note that there are two subtle requirements which are somewhat implicit in the construction. First, for the attack to work, we require that N be a $2^{2\lambda-2}$-bit number. This requirement holds since we assume each prime factor is at least $2^{\lambda-1}$ bits. This requirement is necessary to prevent "overflow" when multiplying by 2 in the attack. Second, we require that $(2^\lambda r + m) \in \mathbb{Z}_N^*$; however, this is assumed to be the case with overwhelming probability.*

Problem 3. A simpler variant of AES encryption?

The Advanced Encryption Standard (AES) is a block cipher that encrypts messages in rounds (typically $r = 10$) using a series of random round keys $k_1, \ldots, k_r$ and a deterministic round function R. At a high level, when encrypting a message using AES, the input to round 1 is the message and the input to the round function R at round i is the output of R from round $i-1$. The round function is deterministic and permutes the input using the random round key k_i. Using the keys $k_1, \ldots, k_r$, it is possible to "unroll" each round and recover the original message (i.e., using the keys it is possible to decrypt and recover the message). However, without knowledge of the round keys, it is conjectured that AES is a pseudo-random permutation. In this problem we will explore a simpler version of the same idea; your job will be to break it.

Consider the following simplification of AES encryption with keys $k_1, \ldots, k_r$ (r is the number of rounds). Define the following *round function* $R(k_i, x) = x^{-1} + k_i \bmod p$ where p is some large prime (e.g., p is a 128 bit prime). (You may assume that $0^{-1} = 0 \pmod p$.) Note that R is public and known to everyone; only the random round keys are kept secret. Define the function $F(k_1, \ldots, k_r, x)$ to output R evaluated iteratively for r rounds (using the round key k_i for round i). That is, on input $(k_1, \ldots, k_r, x)$, set

$$\begin{aligned} x_1 &\leftarrow R(k_1, x), \\ x_2 &\leftarrow R(k_2, x_1), \\ x_3 &\leftarrow R(k_3, x_2), \\ &\ldots, \\ x_r &\leftarrow R(k_r, x_{r-1}), \end{aligned}$$

and output $F(k_1, \ldots, k_r, x) = x_r$.

(a) **Show that F is a permutation. That is, there exists a function F^{-1} such that $F^{-1}(k_1, \ldots, k_r, F(k_1, \ldots, k_r, x)) = x$.**

Solution

The function R_{inv} defined by $R_{\mathsf{inv}}(x) = \begin{cases} 0 & \text{if } x = 0 \\ x^{-1} \bmod p & \text{otherwise} \end{cases}$ is clearly a permutation, since it is inverted by itself. The function $R_{\mathsf{add}}(k_i, \cdot)$ for any $k_i \in \mathbb{Z}_p$ defined by $R_{\mathsf{add}}(k_i, x) = x + k_i \bmod p$ is also clearly a permutation, since it is inverted by $R_{\mathsf{add}}(-k_i, \cdot)$.

Note that each round function $R(k_i, x) = R_{\mathsf{add}}(k_i, R_{\mathsf{inv}}(x))$, i.e. a composition of permutations. Then for any choice of $k_1, \ldots, k_r$, F is a composition of $2r$ permutations, and thus, by the below lemma, a permutation itself.

Lemma 2. *The composition $P : X \to X$ of permutations $P_i : X \to X$ from $i = 1$ to $i = r$ (i.e. $P(x) = P_r(\cdots(P_1(x))))$ is also a permutation.*

Proof. The P_i are permutations, so let $P_i^{-1} : X \to X$ be an inverse for each P_i. Define P^{-1} as the composition of all the P_i^{-1} from $i = r$ to $i = 1$. Then we have $\forall\ x \in X$,

$$P^{-1}(P(x)) = P_1^{-1}(\cdots(P_r^{-1}(P_r(\cdots(P_1(x)))))) = P_1^{-1}(P_1(x)) = x$$

$$P(P^{-1}(x)) = P_r(\cdots(P_1(P_1^{-1}(\cdots(P_r^{-1}(x)))))) = P_r(P_r^{-1}(x)) = x$$

and thus P is a permutation. □

(b) **Provide an attack breaking this toy version of AES. That is, show that F is not a pseudo-random permutation. (Hint: in fact, given $F(k_1, \ldots, k_r, x)$, $F(k_1, \ldots, k_r, y)$, $F(k_1, \ldots, k_r, z)$ for three inputs x, y, z, you can compute $F(k_1, \ldots, k_r, w)$ for any w.)**

Solution

Claim 3. *For random fixed $k = (k_1, \ldots, k_r)$ and an arbitrary nonzero input x, $F(k, x) = x_r = \frac{Ax+B}{Cx+D} \bmod p$ for constants $A, B, C, D \in \mathbb{Z}_p$ except with probability $\leq \frac{r}{p}$.*

Proof. First note that for every round i, k_i is uniform in $\mathbb{Z}_p \implies x_i$ is uniform in $\mathbb{Z}_p \implies \Pr[x_i = 0] = \frac{1}{p}$. Thus by the union bound, $\Pr[x_i = 0 \text{ for some } 1 \leq i \leq r] \leq \frac{r}{p}$. For the rest of the proof we will assume $x_i \neq 0 \ \forall \ 1 \leq i \leq r$.
We will proceed by induction.

Base case. $i = 1$.

$$x_1 = x^{-1} + k_1 = \frac{1}{x} + \frac{k_1 x}{x} = \frac{k_1 x + 1}{x} \bmod p$$

so we have shown our claim for $A = k_1, B = 1, C = 1, D = 0$.

Inductive step. Assume that $x_{i-1} = \frac{ax+b}{cx+d} \bmod p$ for some constants $a, b, c, d \in \mathbb{Z}_p$.

$$x_i = x_{i-1}^{-1} + k_i = \frac{cx+d}{ax+b} + \frac{k_i(ax+b)}{ax+b} = \frac{(c+k_i a)x + (d+k_i b)}{ax+b} \bmod p$$

so we have shown our claim for $A = c + k_i a, B = d + k_i b, C = a, D = b$.

□

Solution (continued...)

Let $\lambda = |p|$ and $r = \mathsf{poly}(\lambda)$, so that $\frac{r}{p}$ is $\mathsf{negl}(\lambda)$. Let x_1, x_2, x_3, x^* be arbitrary distinct nonzero inputs in $\mathbb{Z}_p$. Define $\mathcal{A}$ as follows:

```
Algorithm $\mathcal{A}^{\mathcal{O}}$
for $i \in 1, 2, 3 : \alpha_i \leftarrow \mathcal{O}(x_i)$
if $\alpha_i = \alpha_j$ for some $i \neq j$ : return 1 (“pseudorandom”)
find a vector $(A^*, B^*, C^*, D^*) \in \mathbb{Z}_p^4$ in the null space of $M$ working over $\mathbb{Z}_p$
if none exists, return 0 (“random”)
$\alpha^* \leftarrow \mathcal{O}(x^*)$
if $\alpha^* = \dfrac{A^*x^* + B^*}{C^*x^* + D^*} \bmod p$ : return 1 (“pseudorandom”)
else : return 0 (“random”)
```

where

$$M \leftarrow \begin{bmatrix} x_1 & 1 & -\alpha_1 x_1 & \alpha_1 \\ x_2 & 1 & -\alpha_2 x_2 & \alpha_2 \\ x_3 & 1 & -\alpha_3 x_3 & \alpha_3 \end{bmatrix}.$$

Runtime. $\mathcal{A}$ clearly runs in polynomial time, since it only queries the oracle, checks the result of efficient computations, and performs efficient linear algebra operations.

Correctness. Consider the world where $\mathcal{O} = F(k, \cdot)$. By the above claim, except with negligible probability, we have constants $A, B, C, D \in \mathbb{Z}_p$ such that

$$\alpha_i = F(k, x_i) = \frac{Ax_i + B}{Cx_i + D} \implies \alpha_i(Cx_i + D) = Ax_i + B \implies Ax_i + B - C\alpha_i x_i - D\alpha_i = 0$$

$\implies$ (A, B, C, D) is in the null space of M. By the below lemma, (A^*, B^*, C^*, D^*) is a scalar multiple of (A, B, C, D) mod p, so $\alpha^* = \frac{Ax^*+B}{Cx^*+D} = \frac{A^*x^*+B^*}{C^*x^*+D^*}$ mod p. Therefore $\mathcal{A}^{F(k,\cdot)}$ outputs 1 except with negligible probability.
If R is a truly random function, $\mathcal{A}^R$ outputs 0 except with $\mathsf{negl}(\lambda) + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$ probability: The α_i are all uniformly random, so by a standard birthday paradox argument $\Pr[\alpha_i = \alpha_j \text{ for some } i \neq j] = 1 - \frac{p(p-1)(p-2)}{p^3} = \mathsf{negl}(\lambda)$, and α^* is uniformly random, so given any fixed vector $(A^*, B^*, C^*, D^*) \in \mathbb{Z}_p^4$, $\Pr[\alpha^* = \frac{A^*x^*+B^*}{C^*x^*+D^*}] = \frac{1}{p} = \mathsf{negl}(\lambda)$.

$$\implies \Pr_{K \xleftarrow{R} \mathbb{Z}_p^r}[\mathcal{A}^{G_K}(1^\lambda) = 1] - \Pr_{\text{random } R(\cdot)}[\mathcal{A}^R(1^\lambda) = 1] = (1 - \mathsf{negl}(\lambda)) - \mathsf{negl}(\lambda) = \mathsf{nonnegl}(\lambda)$$

Thus $\mathcal{A}$ successfully distinguishes $F(k, \cdot)$ from a truly random function.

Solution (continued...)

Lemma 4. *M's null space has dimension ≤ 1 over $\mathbb{Z}_p$.*

Proof. Suppose that M has rank < 3 over $\mathbb{Z}_p$. Then we have

$$v(x_1, 1, -\alpha_1 x_1, -\alpha_1) + w(x_2, 1, -\alpha_2 x_2, -\alpha_1) = (x_3, 1, -\alpha_3 x_3, -\alpha_3) \bmod p$$

i.e.

$$\begin{aligned}
&v + w = 1 \implies v = 1 - w \bmod p\\
&vx_1 + (1-v)x_2 = x_3 \bmod p\\
&v\alpha_1 + (1-v)\alpha_2 = \alpha_3 \bmod p\\
&v\alpha_1 x_1 + (1-v)\alpha_2 x_2 = \alpha_3 x_3 \bmod p\\
&\implies [vx_1 + (1-v)x_2] \cdot [v\alpha_1 + (1-v)\alpha_2] = v\alpha_1 x_1 + (1-v)\alpha_2 x_2 = \alpha_3 x_3 \bmod p\\
&\implies (v^2 - v)\alpha_1 x_1 + v(1-v)\alpha_1 x_2 + v(1-v)\alpha_2 x_1 + ((1-v)^2 - (1-v))\alpha_2 x_2 = 0 \bmod p\\
&\implies v(v-1)(x_1 - x_2)(\alpha_1 - \alpha_2) = 0 \bmod p
\end{aligned}$$

p is prime, so one of the following must be true:

$$\begin{aligned}
&v = 0 \implies x_2 = x_3 \implies \text{contradiction}\\
&v - 1 = 0 \implies w = 0 \implies x_1 = x_3 \implies \text{contradiction}\\
&x_1 = x_2 \implies \text{contradiction}\\
&\alpha_1 = \alpha_2 \implies \text{algorithm already returned 1} \implies \text{contradiction}
\end{aligned}$$

Thus M has rank ≥ 3, so it has nullity ≤ 1 over $\mathbb{Z}_p$. □

Problem 4. Circular security

Alice has an arbitrary public key encryption scheme where the messages and keys are strings in $\{0,1\}^\lambda$. Alice generates her keys pk and sk and stores them on her hard drive. Alice then encrypts the whole hard drive using her public key pk. An attacker gains access to the encrypted contents of her hard-drive. Can you argue that her data is still secure? Formally,

Definition 5 (Circular Security)**.** *A public key encryption scheme* $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *with message space* $\mathcal{M} = \mathcal{K}$ *is said to be circular secure if for all PPT* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$\Pr\left[\begin{array}{l} (pk, sk) \leftarrow \mathsf{PKE.Gen}(1^\lambda);\ c^* \leftarrow \mathsf{PKE.Enc}(pk, sk); \\ (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1(pk, c^*); \\ b \overset{R}{\leftarrow} \{0,1\};\ c_b \leftarrow \mathsf{PKE.Enc}(pk, m_b); \\ b' \leftarrow \mathcal{A}_2(pk, c_b, \mathsf{state}) : \\ b' = b \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Does every IND-CPA encryption scheme satisfy Definition 5? Either provide a proof showing all IND-CPA-secure public key encryption schemes satisfy Definition 5 or provide a counterexample.

Solution

We will construct an IND-secure PKE scheme Π' which is not circular secure. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-secure PKE scheme. Let $\Pi' = (\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$ where

$$\mathsf{Enc}'(pk, m) = \begin{cases} 0||m & \text{if } \mathsf{Dec}(m, \mathsf{Enc}(pk, m')) = m' \text{ for } m' \overset{R}{\leftarrow} \{0,1\}^n \\ 1||\mathsf{Enc}(pk, m) & \text{otherwise} \end{cases}$$

and

$$\mathsf{Dec}'(sk, c) = \begin{cases} c_{[2:]} & \text{if } c_1 = 0 \\ \mathsf{Dec}(sk, c_{[2:]}) & \text{if } c_1 = 1. \end{cases}$$

Given any PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$, by the IND-security of Π we have

$$\Pr\left[\begin{array}{l} (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda); \\ (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda, pk); \\ b \overset{R}{\leftarrow} \{0,1\};\ c_b \leftarrow 1 \mid\mid \mathsf{Enc}(pk, m_b); \\ b' \leftarrow \mathcal{A}_2(pk, c_b, \mathsf{state}) \end{array} : \ b' = b\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

(since inserting a 1 at the beginning of every ciphertext leaks no information). Note that

$$\Pr_{pk, sk \leftarrow \mathsf{Gen}}[\mathsf{Enc}'(pk, m_b) \neq 1||\mathsf{Enc}(pk, m_b)] = \Pr_{\substack{pk, sk \leftarrow \mathsf{Gen} \\ m' \overset{R}{\leftarrow} \{0,1\}^n}}[\mathsf{Dec}(m_b, \mathsf{Enc}(pk, m') = m'] \leq \mathsf{negl}(\lambda)$$

by lemma 6, proved below. Then even if $(\mathcal{A}_1, \mathcal{A}_2)$ wins the IND-security game against Π' whenever $\mathsf{Enc}'(pk, m_b) \neq 1 \mid\mid \mathsf{Enc}(pk, m_b)$, its advantage is still $\mathsf{negl}(\lambda)$. Thus Π' is IND-secure.
However, Π' is obviously not circular secure. Given $c^* = \mathsf{Enc}'(pk, sk) = 0||sk$, an adversary can simply extract the secret key and decrypt the challenge ciphertext to win the game against Π' with probability 1.

Solution (continued...)

We now prove the lemma we used above, which essentially states that no PPT adversary can produce a key which has a non-negligible chance of correctly decrypting an encryption of a random message without breaking IND-CPA security.

Lemma 6. *There does not exist a PPT $\mathcal{A}_1$ such that*

$$\Pr\left[\begin{array}{l}(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda);\\ (s_0, s_1) \leftarrow \mathcal{A}_1(pk);\\ b \xleftarrow{R} \{0,1\};\ m' \xleftarrow{R} \{0,1\}^n\end{array} \quad : \quad \mathsf{Dec}(m_b, \mathsf{Enc}(pk, m')) = m' \right] \geq \mathsf{nonnegl}(\lambda).$$

*Proof.*Suppose towards contradiction that such a $\mathcal{A}_1$ exists. Define $(\mathcal{B}_1, \mathcal{B}_2)$ as follows:

```
Algorithm $\mathcal{B}_1(1^\lambda, pk)$
  $m_0, m_1 \xleftarrow{R} \{0,1\}^n$
  return $m_0, m_1, \mathsf{state} = \{\}$
```

```
Algorithm $\mathcal{B}_2(1^\lambda, pk, c_b, \mathsf{state})$
  $s_0, s_1 \leftarrow \mathcal{A}_1(1^\lambda, pk)$
  $b' \xleftarrow{R} \{0,1\}$
  $m \leftarrow \mathsf{Dec}(s_{b'}, c)$
  if $m = m_0$ : return $0$
  if $m = m_1$ : return $1$
  return $\tilde{b} \xleftarrow{R} \{0,1\}$
```

Runtime. $\mathcal{B}_1$ and $\mathcal{B}_2$ clearly run in $\mathsf{poly}(\lambda)$ time, since other than sending and receiving messages it only performs efficient sampling and comparison operations and runs PPT $\mathcal{A}_1$ once.

Correctness. By assumption and uniform distribution of m_0 and m_1, we have

$$\begin{aligned}
&\Pr\left[\begin{array}{l}(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda);\\ (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{B}_1(1^\lambda, pk);\\ b \xleftarrow{R} \{0,1\};\ c_b \leftarrow \mathsf{Enc}(pk, m_b);\\ b' \leftarrow \mathcal{B}_2(1^\lambda, pk, c_b, \mathsf{state})\end{array} \quad : \quad b' = b\right]\\
&= \Big(1 - \Pr[\mathsf{Dec}(s_{b'}, \mathsf{Enc}(pk, m_b)) = m_{\bar{b}}]\Big)\\
&\qquad \cdot \left[\Pr[\mathsf{Dec}(s_{b'}, \mathsf{Enc}(pk, m_b)) = m_b] + \frac{1}{2}(1 - \Pr[\mathsf{Dec}(s_{b'}, \mathsf{Enc}(pk, m_b)) = m_b])\right]\\
&= \left(1 - \frac{1}{2^\lambda}\right)\left(\mathsf{nonnegl}(\lambda) + \frac{1}{2}(1 - \mathsf{nonnegl}(\lambda))\right) = \frac{1}{2} + \mathsf{nonnegl}(\lambda)
\end{aligned}$$

so $(\mathcal{B}_1, \mathcal{B}_2)$ wins the IND-security game against Π with non-negligible advantage, but this is a contradiction, since we know Π is IND-secure. Thus no such $\mathcal{A}_1$ can exist. □

Problem 5. Digital signatures from RSA

Alice wants to use the RSA digital signature scheme defined as follows.

```
Gen(1^λ)
  sample random distinct
        $\lambda$-bit primes $p$ and $q$
  $p' \leftarrow p^{-1} \pmod{q}$
  $q' \leftarrow q^{-1} \pmod{p}$
  $N \leftarrow pq$
  $e \xleftarrow{R} \mathbb{Z}_N^*,\ d \leftarrow e^{-1} \pmod{\phi(N)}$
  $sk := (d, p, q, p', q'),\ pk := (N, e)$
  return $(sk, pk)$
```

```
Sign(pk, sk, m ∈ M)
  parse $pk = (N, e)$
        and $sk = (d, p, q, p', q')$
  $\sigma_p \leftarrow m^d \pmod{p}$
  $\sigma_q \leftarrow m^d \pmod{q}$
  $\sigma \leftarrow \sigma_p q q' + \sigma_q p p' \pmod{N}$
  return $\sigma$
```

```
Verify(pk, m, σ)
  parse $pk = (N, e)$
  if $\sigma^e = m \pmod{N}$ :
    return 1
  else : return 0
```

Note that in Sign*, the signature is computed by applying the Chinese remainder theorem. This is done so as to make the signature algorithm more efficient as it results in two exponentiations modulo a λ-bit number (and one addition mod N) rather than a single exponentiation mod N. This does not compromise security and is a standard way of improving efficiency in practice.*

Unfortunately, the part of Alice's computer responsible for modular exponentiation is being affected by cosmic radiation, so anytime she computes a signature, each bit of σ_p and each bit of σ_q is flipped with probability $\frac{1}{\lambda}$. (Yes, this is a real phenomenon.[1]) The computation of σ from σ_p and σ_q is unaffected by radiation.

> ~~**Given the problems with Alice's computer, show that she cannot expect security of this signature scheme to hold. That is, provide an attack on the EUF-CMA security of this scheme, noting that the Sign oracle in the EUF-CMA security game will suffer from the bit-flipping described above.**~~
>
> **Show that if Alice outputs even a single corrupted signature, Eve can use it to factor N and recover Alice's secret key with non-negligible probability.**

[1] https://www.wnycstudios.org/podcasts/radiolab/articles/bit-flip

Solution

We will first look at how we can factor N when given a signature corrupted in a certain way. Then, we will compute the probability of a corruption occurring in this way to show that we can factor N with non-negligible probability.

Let $m \in \mathbb{Z}_N^*$ be any message. Suppose (without loss of generality) that the computation of σ is corrupted so $\sigma_p = m^d$ while $\sigma_q \neq m^d$ (the case where σ_p is corrupted is symmetric). Then, we can compute

$$p = \mathsf{gcd}((\sigma^e - m) \bmod N, N).$$

In other words, we recover the prime factor of N. This follows because we have that

$$\begin{aligned} \sigma^e &= m \pmod p \\ \sigma^e &\neq m \pmod q. \end{aligned}$$

Therefore, we have that

$$\begin{aligned} \sigma^e - m &= 0 \pmod p \\ \sigma^e - m &\neq 0 \pmod q. \end{aligned}$$

which implies that p divides both $\sigma^e - m$ and N. (Note that when the signature is **not** corrupted, then N divides both $\sigma^e - m$ and N so the GCD returns N rather than p.)

Probability analysis We now compute the probability that either σ_p or σ_q is corrupted (but not both!). The probability that σ_p [resp. σ_q] has no bits flipped is:

$$(1 - \frac{1}{\lambda})^\lambda \approx \frac{1}{e} \text{ (asymptotically)},$$

and likewise, the probability that σ_p [resp. σ_q] has at least one bit flipped is:

$$1 - (1 - \frac{1}{\lambda})^\lambda \approx 1 - \frac{1}{e} \text{ (asymptotically)}.$$

Because these events are independent, we get that the following probability for one signature being corrupted while the other is not:

$$(1 - \frac{1}{e})(\frac{1}{e}) \approx 0.232,$$

which is non-negligible.

Problem 6. Fun with collision-resistant hash functions

A hash function family $\mathcal{H}$ is a pair of algorithms $(\mathsf{Gen}, \{h_k\}_k)$ where:

1. $\mathsf{Gen}(1^\lambda)$ is an efficient randomized algorithm that outputs a key k.
2. $h_k : \{0,1\}^* \to \{0,1\}^\lambda$ is an efficient algorithm parameterized by key k that compresses arbitrary length strings to λ-bit strings.

A collision-resistant hash function (CRHF) is defined as follows.

Definition 7 (Collision-resistant hash function)**.** *A hash function family $\mathcal{H} = (\mathsf{Gen}, \{h_k\}_k)$ is said to be collision-resistant if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that:*

$$\Pr\left[\begin{array}{l} k \leftarrow \mathsf{Gen}(1^\lambda); \\ (x, x') \leftarrow \mathcal{A}(1^\lambda, k, \mathcal{H}) : x \neq x' \wedge h_k(x) = h_k(x') \end{array}\right] \leq \mathsf{negl}(\lambda).$$

That is, an adversary $\mathcal{A}$ that is given the hash function *and key* cannot generate a collision (i.e., two values $x \neq x'$ such that $h_k(x) = h_k(x')$) except with negligible probability $\mathsf{negl}(\lambda)$.

Let $\mathcal{H}_1 = (\mathsf{Gen}_1, \{h^1_k\}_k)$ and $\mathcal{H}_2 = (\mathsf{Gen}_2, \{h^2_k\}_k)$ be two CRHF families. Define $\mathsf{Gen}'(1^\lambda)$ to return (k_1, k_2) where $k_1 \leftarrow \mathsf{Gen}_1(1^\lambda), k_2 \leftarrow \mathsf{Gen}_2(1^\lambda)$. For each of the following, either prove that $\mathcal{H}' = \left(\mathsf{Gen}', \{h'_{k_1,k_2}\}_{k_1,k_2}\right)$ is a CRHF family or provide a counterexample showing that it is not always collision-resistant.

(a) $h'_{k_1,k_2}(x) = h^1_{k_1}(x) \mid\mid h^2_{k_2}(x)$

Solution

$\mathcal{H}'$ is collision resistant. We claim that each collision for $\mathcal{H}'$ immediately gives a collision for $\mathcal{H}_1$ (the case for $\mathcal{H}_2$ is symmetric), therefore no adversary can produce collisions for $\mathcal{H}'$ with non-negligible probability.

We will formally prove this with a reduction. Suppose, towards contradiction, that $\mathcal{H}'$ was not collision-resistant. Then, there exists a PPT adversary $\mathcal{A}$ that produces a collision for $\mathcal{H}'$ with non-negligible probability $\delta(\lambda)$. Construct PPT adversary $\mathcal{B}$ for $\mathcal{H}_1$ as follows.

```
Algorithm $\mathcal{B}(1^\lambda, k_1)$
$k_2 \leftarrow \mathsf{Gen}_2(1^\lambda)$
$(x, x') \leftarrow \mathcal{A}(1^\lambda, k_1, k_2)$
return $(x, x')$
```

Analysis: If $\mathcal{A}$ succeeds, then $h^1_{k_1}(x)||h^2_{k_2}(x) = h^1_{k_1}(x')||h^2_{k_2}(x')$ by definition of finding a collision. This immediately implies that $h^2_{k_2}(x) = h^2_{k_2}(x')$ and so $\mathcal{B}$ succeeds in finding a collision for $\mathcal{H}_2$ whenever $\mathcal{A}$ succeeds. As such, $\mathcal{H}'$ is a collision-resistant hash function.

(b) $h'_{k_1,k_2}(x) = h^1_{k_1}(h^2_{k_2}(x))$

Solution

$\mathcal{H}'$ is collision resistant.

Every collision on $\mathcal{H}'$ is of the form $h^1_{k_1}(h^2_{k_2}(x)) = h^1_{k_1}(h^2_{k_2}(x'))$. Then, there are two cases to consider:

Case 1: $h^2_{k_2}(x) = h^2_{k_2}(x')$. In this first case, we see that (x, x') where $x \neq x'$ is a collision for $h^2_{k_2}$. Thus a collision is found for $\mathcal{H}_2$.

Case 2: $h^2_{k_2}(x) \neq h^2_{k_2}(x')$. In this second case, we see that (y, y') where $y = h^2_{k_2}(x)$ and $y' = h^2_{k_2}(x')$ is a collision for $\mathcal{H}_1$.

In both cases we obtain a collision either on $\mathcal{H}_1$ or $\mathcal{H}_2$. Hence, whenever a collision is found for $\mathcal{H}'$, we necessarily also get a collision for $\mathcal{H}_1$ or $\mathcal{H}_2$. Therefore if there exists a PPT adversary $\mathcal{A}$ that can find collisions for $\mathcal{H}'$ with non-negligible probability, then there also exists an adversary $\mathcal{B}$ that can find collisions for $\mathcal{H}_1$ or $\mathcal{H}_2$, which contradicts the collision resistance assumption for those two hash families.