Cryptography—Homework 1
Solutions and Scores

Sapienza University of Rome
Master Degree in Computer Science

Daniele Venturi

November 18, 2016

The grades are shown on Tab. 1 on page 10.

1 Perfect Secrecy and One-Time Pad — 20 Points

(a) Note that whenever the all-zero key is chosen in the one-time pad, we obtain $\mathsf{Enc}(k, m) = 0^\ell \oplus m = m$. Is this a problem? In particular, suppose to modify the one-time pad by requiring that the key is sampled from the set $\mathcal{K}' := \{0,1\}^\ell \setminus \{0^\ell\}$. Is the resulting encryption scheme still perfectly secure? Prove your answer.

Solution: The modified scheme does not meet perfect secrecy, as now $|\mathcal{K}'| = 2^\ell - 1 < 2^\ell = |\mathcal{M}|$, and we know that $(\mathsf{Enc}, \mathsf{Dec})$ is perfectly secret if and only if $|\mathcal{K}| = |\mathcal{M}|$. The latter can be also seen directly by taking M to be uniform over $\{0,1\}^\ell$; for any possible $m \in \{0,1\}^\ell$ we get:

$$\Pr[M = m | C = m] = 0 \neq 2^{-\ell} = \Pr[M = m],$$

which contradicts perfect secrecy.

We conclude that we should not exclude the zero key 0^ℓ, and there is no problem at all if such a key would be chosen. The point is that the adversary has no way of knowing the key is 0^ℓ, so the fact that the ciphertext is equal to the plaintext in this case does not help the adversary.

(b) Let $(\mathsf{Enc}, \mathsf{Dec})$ be a perfectly secret encryption scheme. Refute the following statement: For all distributions M over the message space $\mathcal{M}$, for all $m, m' \in \mathcal{M}$, for all $c \in \mathcal{C}$, we have:

$$\Pr[M = m | C = c] = \Pr\left[M = m' | C = c\right].$$

(**Hint:** Let $\mathcal{M} = \{m_0, m_1\}$ and consider the distribution M such that $\Pr[M = m_0] = 3/4$ and $\Pr[M = m_1] = 1/4$.)

Solution: As hinted above, consider $\mathcal{M} = \{m_0, m_1\}$ with the distribution M such that $\Pr[M = m_0] = 3/4$ and $\Pr[M = m_1] = 1/4$. By definition of perfect secrecy, for all $c \in \mathcal{C}$, we get:

$$\Pr[M = m_0 | C = c] = \Pr[M = m_0] \neq \Pr[M = m_1] = \Pr[M = m_1 | C = c],$$

showing that the statement of the exercise does not hold.

(c) Assume we want to use the one-time pad as a deterministic MAC in the following natural way: Define $\mathsf{Mac}(k, m) = m \oplus k = \phi$, where $\mathcal{K} = \mathcal{M} = \Phi = \{0,1\}^n$. Show that this MAC is not one-time statistically secure.

Solution: Given a valid pair (m, ϕ) such that $\phi = m \oplus k$ consider the adversary that outputs (m^*, ϕ^*) where for any $\delta \in \{0,1\}^n \setminus \{0^n\}$ we define $m^* = m \oplus \delta$ and $\phi^* = \phi \oplus \delta$. Clearly, $m^* \neq m$ and moreover $\phi^* = (m \oplus \delta) \oplus k = m^* \oplus k$ is a valid pair, contradicting one-time statistical security of the MAC.

2 Universal Hashing — 10 Points

(a) A family of functions $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ is called *universal* if for all distinct inputs $x, x' \in \mathcal{X}$ we have:

$$\Pr\left[h_s(x) = h_s(x') : \ s \leftarrow\!\!\$\ \mathcal{S}\right] \leq |\mathcal{Y}|^{-1}.$$

Show that any family $\mathcal{H}$ that is pairwise independent (as defined in class) is also universal.

Solution: This exercise had a small typo, in that the definition of universality and pairwise independence need to be considered with equalities. The correct definition is as follows: The family $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ is called *universal* if for all distinct inputs $x, x' \in \mathcal{X}$ we have:

$$\Pr_{s \leftarrow\!\!\$\ \mathcal{S}}\left[h_s(x) = h_s(x')\right] = \frac{1}{|\mathcal{Y}|},$$

On the other hand, as defined in class, the family $\mathcal{H}$ is called *pairwise independent* if for all distinct $x, x' \in \mathcal{X}$ and all y, y' we have:

$$\Pr_{s \leftarrow\!\!\$\ \mathcal{S}}\left[h_s(x) = y \wedge h_s(x') = y'\right] = \frac{1}{|\mathcal{Y}|^2}.$$

A straightforward calculation then shows that:

$$\Pr_{s \leftarrow\!\!\$\ \mathcal{S}}\left[h_s(x) = h_s(x')\right] = \sum_{y \in \mathcal{Y}} \Pr_{s \leftarrow\!\!\$\ \mathcal{S}}\left[h_s(x) = y \wedge h_s(x') = y\right] = |\mathcal{Y}| \cdot \frac{1}{|\mathcal{Y}|^2} = \frac{1}{|\mathcal{Y}|},$$

as desired.

(b) Let $\mathbb{F}$ be a finite field, and consider the following family of hash functions $\mathcal{H}$ with $\mathcal{Y} = \mathbb{F}$ and $\mathcal{X} = \mathcal{S} = \mathbb{F}^t$ for some value $t \in \mathbb{N}$. For a secret key $s = (s_1, \ldots, s_t) \in \mathbb{F}^t$, and input $x = (x_1, \ldots, x_t) \in \mathbb{F}^t$, define $h_s(x) := \sum_{i=1}^{t} s_i \cdot x_i$ where all operations take place in $\mathbb{F}$. Show that $\mathcal{H}$ is universal.

Solution: Let $x = (x_1, \ldots, x_t)$ and $x' = (x'_1, \ldots, x'_t)$, and define $\delta_i = x'_i - x_i$ for all $i \in [t]$. If x and x' are distinct, at least one of the δ_i's is different from zero, say this happens for $\delta_1 = x_1 - x'_1$. We need to compute the probability that $h_s(x) = h_s(x')$ over the choice of the key $s \in \mathbb{F}^t$. But notice that,

$$h_s(x) = h_s(x') \Leftrightarrow \sum_{i=1}^{t} s_i \cdot x_i = \sum_{i=1}^{t} s_i \cdot x'_i \Leftrightarrow s_1 \cdot \delta_1 = -\sum_{i=1}^{t} s_i \cdot \delta_i \Leftrightarrow s_1 = \frac{-\sum_{i=1}^{t} s_i \cdot \delta_i}{\delta_1}.$$

The above quantity is well defined, since $\delta_1 \neq 0$; we conclude

$$\Pr_{s \leftarrow\$ \mathbb{F}^t}\left[h_s(x) = h_s(x')\right] = \frac{1}{|\mathbb{F}|}.$$

3 One-Way Functions — 20 Points

(a) Let $f : \{0,1\}^n \to \{0,1\}^n$ be a candidate OWF. Show that there always exists an inefficient attacker $\mathcal{A}_1$ inverting the function with probability one, and an efficient attacker $\mathcal{A}_2$ inverting the function with probability 2^{-n}.

Solution: Consider the attacker $\mathcal{A}_1$ that given $y = f(x)$ for random $x \leftarrow\$ \{0,1\}^n$ tries all possible $x' \in \{0,1\}^n$, computes $f(x') = y'$, and outputs x' such that $y' = y$. Clearly, x' must exist as $y = f(x)$ for some $x \in \{0,1\}^n$. Hence, $\mathcal{A}_1$ succeeds with probability 1 but runs in exponential time (in n).

Similarly, consider the attacker $\mathcal{A}_2$ that simply outputs a random $x' \in \{0,1\}^n$; the probability that $x' = x$ is 2^{-n} that is also a bound on the adversary's advantage.

(b) Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $f(x, y) = x + y$, where $|x| = |y|$ and x, y are interpreted as natural numbers.
Solution: This candidate *does not yield* a OWF. Given $f(x, y) = z$, for random x, y, simply return $(z, 0)$. Clearly, $f(z, 0) = z + 0 = x + y = f(x, y)$.

(ii) $g(x_1, x_2) = (f(x_1), x_2)$, where f is a OWF and $|x_1| = |x_2|$.

Solution: This candidate *does yield* a OWF. Let $|x_1| = |x_2| = n(\lambda)$. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x_1', x_2') = \hat{y} : \begin{array}{c} x_1, x_2 \leftarrow\!\!\$\ \{0,1\}^{n(\lambda)}; \hat{y} = g(x_1, x_2) \\ (x_1', x_2') \leftarrow \mathcal{A}(1^\lambda, y) \end{array}\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, pick random $x_2 \in \{0,1\}^n$, run $(x_1', x_2') \leftarrow \mathcal{A}(1^\lambda, (y, x_2))$, and return x_1'. Notice that the input (y, x_2) that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $(y, x_2) = (f(x), x_2)$ for random $x, x_2 \in \{0,1\}^n$. We conclude that $\mathcal{A}'$ inverts f with the same probability that $\mathcal{A}$ inverts g (i.e., with non-negligible probability). This concludes the proof.

(iii) $g(x) = (f(x), f(f(x)))$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.

Solution: This candidate *does yield* a OWF. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x') = \hat{y} :\ x \leftarrow\!\!\$\ \{0,1\}^n; \hat{y} = g(x); x' \leftarrow \mathcal{A}(1^\lambda, y)\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, compute $z = f(y)$, run $x' \leftarrow \mathcal{A}(1^\lambda, (y, z))$, and return x'. Notice that the input (y, z) that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $(y, z) = (f(x), f(f(x)))$ for random $x \in \{0,1\}^n$. We conclude that $\mathcal{A}'$ inverts f with the same probability that $\mathcal{A}$ inverts g (i.e., with non-negligible probability). This concludes the proof.

(iv) $g(x) = f(x||0)$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.

Solution: This candidate *does yield* a OWF. By contradiction, assume that there exists an adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many values of $\lambda \in \mathbb{N}$

$$\Pr\left[g(x') = \hat{y} :\ x \leftarrow\!\!\$\ \{0,1\}^{n-1}; \hat{y} = g(x); x' \leftarrow \mathcal{A}(1^\lambda, y)\right] \geq 1/p(\lambda).$$

Consider the following attacker $\mathcal{A}'$ for inverting f: Upon input $y = f(x)$ for random $x \in \{0,1\}^n$, run $x' \leftarrow \mathcal{A}(1^\lambda, y)$, and return $x'||0$. Notice that as long as the first bit of x is zero, which happens with probability $1/2$, the input y that $\mathcal{A}'$ passes to $\mathcal{A}$ has the distribution $\mathcal{A}$ expects, since $y = f(x''||0))$ for random $x'' \in \{0,1\}^{n-1}$. We conclude that $\mathcal{A}'$ inverts f with probability at least $p(\lambda)/2$ which is non-negligible. This concludes the proof.

4 Pseudorandom Generators 20 Points

(a) Show that no PRG can be secure against computationally unbounded adversaries. In particular, let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling PRG. Show that there is an exponential-time distinguisher breaking the PRG with probability almost one.

Solution: Consider the following exponential time distinguisher $\mathcal{D}$. Upon input $y \in \{0,1\}^{2\lambda}$ the goal of $\mathcal{D}$ is to distinguish whether $y = G(s)$ for a random $s \leftarrow\!\!\$\ \{0,1\}^\lambda$, or $y \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$. The distinguisher simply outputs 1 if and only if there exists some $s \in \{0,1\}^\lambda$ such that $y = G(s)$; note that this computation is performed by computing $G(s)$ for all possible seeds $s \in \{0,1\}^\lambda$.

Now, if y really comes from the range of the PRG, $\mathcal{D}$ outputs 1 with probability 1. On the other hand, assume that $y \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$. Then with probability $2^{-\lambda}$ the value y is actually outside the range of the PRG; this is because G receives an input of length λ and thus its range consists of at most 2^λ values, whereas y is sampled from a set of $2^{2\lambda}$ possible values, and $2^\lambda/2^{2\lambda} = 2^{-\lambda}$. We conclude that,

$$\left|\Pr\left[\mathcal{D}(y) = 1 :\ s \leftarrow\!\!\$\ \{0,1\}^\lambda; y = G(s)\right] - \Pr\left[\mathcal{D}(y) = 1 :\ y \leftarrow\!\!\$\ \{0,1\}^{2\lambda}\right]\right| \geq 1 - 2^{-\lambda},$$

and thus $\mathcal{D}$ distinguishes with overwhelming probability.

(b) Analyze the following candidate PRGs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $G'(s) = G(s_1)||\ldots||G(s_n)$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and $s = s_1||\ldots||s_n \in \{0,1\}^{\lambda n}$.
Solution: This candidate *does yield* a PRG. Fix some polynomial $n := n(\lambda)$. Define the following hybrid distribution:

$$\mathbf{H}_i(\lambda) = (\underbrace{U_{\lambda+\ell}, \ldots, U_{\lambda+\ell}}_{i \text{ times}}, \underbrace{G(U_\lambda), \ldots, G(U_\lambda)}_{n-i \text{ times}}).$$

Notice that $\mathbf{H}_0(\lambda) \equiv G'(U_{\lambda n})$, whereas $\mathbf{H}_n(\lambda) \equiv U_{(\lambda+\ell)n}$. We prove security by a standard hybrid argument. Assume that there exists a distinguisher $\mathcal{D}$, an index $i \in [n]$, and a polynomial $p(\cdot)$ such that, for infinitely many values of $\lambda \in \mathbb{N}$:

$$|\Pr[\mathcal{D}(\mathbf{H}_i(\lambda)) = 1] - \Pr[\mathcal{D}(\mathbf{H}_{i+1}(\lambda)) = 1]| \geq 1/p(\lambda).$$

We construct a distinguisher $\mathcal{D}'$ breaking security of the underlying PRG G. The distinguisher $\mathcal{D}'$, upon input a target value y that is either sampled from $G(U_\lambda)$ or from $U_{\lambda+\ell}$, proceeds as follows:

1. Sample $y_1, \ldots, y_i \leftarrow\!\!\$\ \{0,1\}^{\lambda+\ell}$.

2. Sample $s_{i+2}, \ldots, s_n \leftarrow\!\!\$\ \{0,1\}^\lambda$ and let $y_j = G(s_j)$ for all $i+2 \leq j \leq n$.
3. Return the same as $\mathcal{D}(y_1, \ldots, y_i, y, y_{i+2}, \ldots, y_n)$.

For the analysis, it suffices to note that in the above reduction the input to $\mathcal{D}$ either comes from the distribution $\mathbf{H}_i(\lambda)) = 1$ (in case $y = G(s)$ comes from the PRG), or from the distribution $\mathbf{H}_{i+1}(\lambda)) = 1$ (in case y is random). We conclude that $G'(U_{\lambda n}) \equiv \mathbf{H}_0(\lambda) \approx_c \ldots \approx_c \mathbf{H}_n(\lambda) \equiv U_{(\lambda+\ell)n}$, as desired.

(ii) $G'(s) = G(s||0^{|s|})$, where $G : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda+\ell}$ is a PRG.

(**Hint:** Consider the contrived PRG $G(s)$ that ignores the first half of its input, and returns $\hat{G}(s_{\lambda+1}, \ldots, s_{2\lambda})$ where $\hat{G}$ is itself a PRG stretching λ bits into $2\lambda + \ell$ bits. Argue that G is a PRG, but G' as defined in the exercise is not.)

Solution: As hinted above, let $\hat{G}$ be a PRG stretching λ bits into $2\lambda + \ell$ bits, and consider the PRG $G(s)$ that ignores the first half of its input, and returns $\hat{G}(s_{\lambda+1}, \ldots, s_{2\lambda})$. We argue that G is still a PRG, but G' is not.

The second part of the statement is easy, indeed: $G'(s) = G(s||0^{|s|}) = \hat{G}(0^\lambda)$ and it is trivial to distinguish the outout of G' from random. It remains to prove that G is a PRG, but this is also easy to show. In fact, for any efficient $\mathcal{D}$, we have

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}(y) = 1 :\ s \leftarrow\!\!\$\ \{0,1\}^{2\lambda}; y = G(s)\right] - \Pr\left[\mathcal{D}(y) = 1 :\ y \leftarrow\!\!\$\ \{0,1\}^{2\lambda+\ell}\right]\right| \\
=&\left|\Pr\left[\mathcal{D}(y) = 1 :\ s \leftarrow\!\!\$\ \{0,1\}^{\lambda}; y = \hat{G}(s)\right] - \Pr\left[\mathcal{D}(y) = 1 :\ y \leftarrow\!\!\$\ \{0,1\}^{2\lambda+\ell}\right]\right|
\end{aligned}$$

and the latter quantity is negligible by the fact that $\hat{G}$ is a PRG.

5 Pseudorandom Functions — 10 Points

Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $F_k(x) = G'(k) \oplus x$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and G' denotes the output of G truncated to λ bits.

Solution: This candidate *does not yield* a PRF. Let $\mathcal{D}$ be a distinguisher that is given oracle access to either $F_k(\cdot)$ for a random $k \in \{0,1\}^\lambda$, or to a truly random function $R : \{0,1\}^\lambda \to \{0,1\}^\lambda$. The distinguisher queries its oracle upon input any two distinct values $x, x' \in \{0,1\}^\lambda$, receiving back outputs y, y'. Hence, $\mathcal{D}$ returns 1 if and only if $x \oplus x' = y \oplus y'$.

Clearly, if $\mathcal{D}$'s oracle is $F_k(\cdot)$ (for a uniform $k \leftarrow\!\!\$\ \{0,1\}^\lambda$):

$$y \oplus y' = (G'(k) \oplus x) \oplus (G'(k) \oplus x') = x \oplus x',$$

and thus $\mathcal{D}$ always outputs 1 in such a case. On the other hand, if $\mathcal{D}$'s oracle is R, the probability (over the choice of the function) that $x \oplus x' = R(x) \oplus R(x')$ is equal to $2^{-\lambda}$. Hence, $\mathcal{D}$ has distinguishing advantage nearly equal to 1.

(ii) $F_k(x) := F_x(k)$, where $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a PRF.

Solution: This candidate *does not yield* a PRF. Consider the contrived PRF F that upon input the all-zero key 0^λ always returns 0^ℓ independently of the input, i.e., $F_{0^\lambda}(x) = 0^\ell$ for all $x \in \{0,1\}^\lambda$. On the one hand, it is easy to prove that the above function still defines a PRF (as the "bad key" 0^λ is chosen only with negligible probability). On the other hand, it is easy to distinguish $F_x(k)$ (for a randomly chosen "key" $k \leftarrow\!\!\$\ \{0,1\}^\lambda$) from a truly random function by simply querying input $x = 0^\lambda$ to the target oracle: If the oracle implements the PRF, we will obtain 0^ℓ with probability 1, whereas if the oracle implements a truly random function $R : \{0,1\}^\lambda \to \{0,1\}^\ell$ the value 0^ℓ will be obtained only with probability $2^{-\ell}$.

(iii) $F'_k(x) = F_k(x||0)||F_k(x||1)$, where $x \in \{0,1\}^{n-1}$.

Solution: This candidate *does yield* a PRF. By contradiction, assume there exists an efficient distinguisher $\mathcal{D}'$ that can distinguish (oracle access to) $F'_k(\cdot)$ with a random key from a truly random function $R' : \{0,1\}^{n-1} \to \{0,1\}^n$ with non-negligible probability.

Consider the following distinguisher $\mathcal{D}$, whose goal is to distinguish (oracle access to) $F_k(\cdot)$ with a random key from a truly random function $R : \{0,1\}^n \to \{0,1\}^n$: (1) Upon input an oracle query $x' \in \{0,1\}^{n-1}$ from $\mathcal{D}$, define $x^0 := x'||0$ and $x^1 := x'||1$, query x^0, x^1 to the target oracle receiving back values y^0, y^1, and return $y^0||y^1$ to $\mathcal{D}'$; (2) Output the same as $\mathcal{D}'$. Clearly, $\mathcal{D}$ is roughly as efficient as $\mathcal{D}'$ (in fact, if $\mathcal{D}'$ makes q' oracle queries $\mathcal{D}$ needs to make $q = 2q'$ oracle queries), moreover it perfectly simulates the view of $\mathcal{D}'$ and thus it retains the same (non-negligible) advantage. This finishes the proof.

6 Secret-Key Encryption — 20 Points

Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be a SKE scheme with key space $\mathcal{K}$. Consider the following variant of computational one-time security: For all PPT adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon : \mathbb{N} \to \mathbb{R}$ such that

$$\Pr\left[b = b' : \begin{array}{c} k \leftarrow\!\!\$\ \mathcal{K}; b \leftarrow\!\!\$\ \{0,1\}; (m_0, m_1) \leftarrow \mathcal{A}(1^\lambda) \\ c = \mathsf{Enc}(k, m_b); b' \leftarrow \mathcal{A}(1^\lambda, c) \end{array}\right] \leq \frac{1}{2} + \varepsilon(\lambda), \tag{1}$$

with $|m_0| = |m_1|$, and where the probability is taken over the random choice of k, b, and over the randomness of the algorithm $\mathcal{A}$. Prove that this variant is equivalent to the notion

we defined in class. (This means that the above formulation implies the one we gave in class, and viceversa.)

Solution: For $b \in \{0,1\}$, denote with $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$ the game we considered in class in the definition of one-time computational security for SKE:

Game $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$:

1. $k \leftarrow\!\!\$\ \mathcal{K}$
2. $(m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
3. $c = \mathsf{Enc}(k, m_b)$
4. $b' \leftarrow \mathcal{A}(1^\lambda, c)$

Our original formulation was that Π is one-time computationally secure if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\varepsilon : \mathbb{N} \to \mathbb{R}$ such that

$$\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \varepsilon(\lambda). \tag{2}$$

Let $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda)$ be identical to $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b)$ except that the bit b is not fixed anymore, but it is instead chosen uniformely at random at the beginning of the game, and furthermore the output of $\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda)$ is defined to be 1 if and only if $b' = b$. Clearly, Eq. (1) is equivalent to

$$\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{2} + \varepsilon(\lambda). \tag{3}$$

We first prove that Eq. (2) implies Eq. (3). A simple calculation shows:

$$\begin{aligned}
\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] &= \Pr_{b \leftarrow\!\!\$\ \{0,1\}}\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, b) = b\right] \\
&= \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 0\right] + \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] \\
&= \frac{1}{2}\left(1 - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) + \frac{1}{2}\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] \\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) \qquad (4) \\
&\leq \frac{1}{2} + \frac{1}{2}\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \\
&\leq \frac{1}{2} + \frac{1}{2}\varepsilon(\lambda),
\end{aligned}$$

where the last inequality follows by Eq. (2). Since $\varepsilon(\lambda)$ is negligible, so is $\varepsilon(\lambda)/2$, which proves Eq. (3).

Next, we prove that Eq. (3) implies Eq. (2). Combining Eq. (3) with Eq. (4), we can write:

$$\frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) = \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \leq \varepsilon(\lambda). \tag{5}$$

For any adversary $\mathcal{A}$ consider the adversary $\tilde{\mathcal{A}}$ that outputs the complement of $\mathcal{A}$ (i.e., if $\mathcal{A}$ outputs b', we get that $\tilde{\mathcal{A}}$ outputs $\tilde{b} = 1 - b'$). A calculation similar to the one above shows:

$$\begin{aligned}
\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\tilde{\mathcal{A}}}(\lambda) = 1\right] &= 1 - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \\
&= 1 - \frac{1}{2} - \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right]\right) \\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right). \qquad (6)
\end{aligned}$$

Combining Eq. (3) with Eq. (6), we can write:

$$\frac{1}{2}\left(\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right) = \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\tilde{\mathcal{A}}}(\lambda) = 1\right] - \frac{1}{2} \leq \tilde{\varepsilon}(\lambda), \quad (7)$$

for some negligible function $\tilde{\varepsilon} : \mathbb{N} \to [0, 1]$.

Putting together Eq. (5) and Eq. (7) we obtain:

$$\left|\Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathbf{G}^{\text{1-time}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq 2\varepsilon^*(\lambda),$$

where $\varepsilon^* := \max\{\varepsilon, \tilde{\varepsilon}\}$. Since both $\varepsilon(\lambda)$ and $\tilde{\varepsilon}(\lambda)$ are negligible, so is $\varepsilon^*(\lambda)$, which proves Eq. (2).

Table 1: Grades for the first homework. I've decided to leave the last exercise as optional, since almost nobody got it correctly. The final grade is shown in the last column and it can be obtained from the previous column by multiplying for a normalizing factor of 3/8.

Student ID	**Ex. 1 (20 pt)**	**Ex. 2 (10 pt)**	**Ex. 3 (20 pt)**	**Ex. 4 (20 pt)**	**Ex. 5 (10 pt)**	**Ex. 6 (20 pt)**	**Tot. (80 pt)**	**Tot. (30)**
1734634	14	6	8	9	3	—	40	**15**
Atakishiyev	12	5	7	18	8	—	50	**19**
1288895	20	4	12	15	5	—	56	**21**
1496066	20	10	12	14	7	—	63	**24**
1529821	20	5	10	16	3	—	54	**20**
1532426	20	9	16	16	8	—	69	**26**
1346443	18	8	12	16	3	—	57	**21**
1571294	18	3	12	12	2	—	47	**18**
Guastamacchia	20	10	12	10	3	—	55	**21**
1532831	18	8	14	18	7	—	65	**24**
1603064	18	10	20	20	10	—	78	**29**
1543819	17	10	15	16	5	—	63	**24**
1496408	18	10	10	19	5	—	62	**23**
1503936	20	10	12	15	6	—	63	**24**
Majith	17	10	11	20	6	—	64	**24**
1607862	20	10	20	20	10	—	80	**30 e lode**
1152419	18	10	19	17	3	—	67	**25**
Tuccinardi	20	10	16	11	4	—	61	**23**

Cryptography—Homework 2

Sapienza University of Rome
Master Degree in Computer Science

Daniele Venturi

Due Date: December 19, 2016

1 Luby-Rackoff — 20 Points

Recall that the Feistel permutation $\Psi_f : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined by $\Psi_f(X) := (R, L \oplus f(R))$, where $X = L||R$, with both $L, R \in \{0,1\}^n$, and $f : \{0,1\}^n \to \{0,1\}^n$ is a function. Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^\lambda}$ be a PRF family. An r-round Feistel network $\Psi_{\mathcal{F}}[r]$ is the concatenation of r Feistel permutations, each using an independent PRF $F_{k_i} : \{0,1\}^n \to \{0,1\}^n$ from the family $\mathcal{F}$. More precisely, for every choice of the keys $k_1, \ldots, k_r \leftarrow\!\!\$\ \{0,1\}^n$, a permutation $\Psi_{F_{k_1},\ldots,F_{k_r}}$ in $\Psi_{\mathcal{F}}[r]$ is defined as follows:

$$\Psi_{F_{k_1},\ldots,F_{k_r}}(X) := \Psi_{F_{k_r}}(\cdots \Psi_{F_{k_2}}(\Psi_{F_{k_1}}(X))\cdots).$$

Answer the following questions:

(a) Show that $\Psi_{F_{k_1},\ldots,F_{k_r}}$ can be efficiently inverted for any $r \in \mathbb{N}$.

(b) Show that a $\Psi_{\mathcal{F}}[1]$ is not a PRP family.

(c) Show that a $\Psi_{\mathcal{F}}[2]$ is not a PRP family.

(d) Show that a $\Psi_{\mathcal{F}}[3]$ is not a *strong* PRP family.[1]

[1] A family of permutations $\mathcal{P} = \{P_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^\lambda}$ is called a strong PRP if for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that the following holds:

$$\left|\Pr\left[\mathcal{D}^{P(k,\cdot),P^{-1}(k,\cdot)}(1^\lambda) = 1 :\ k \leftarrow\!\!\$\ \{0,1\}^\lambda\right] - \Pr\left[\mathcal{D}^{R(\cdot),R^{-1}(\cdot)}(1^\lambda) = 1 :\ R \leftarrow\!\!\$\ \$_n\right]\right| \leq \nu(\lambda),$$

where $\$_n$ is the set of all possible permutations over $\{0,1\}^n$.

2 Message Authentication — 20 Points

(a) Consider a variant of the game defining UF-CMA security of a MAC $\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$, with key space $\mathcal{K} = \{0,1\}^\lambda$, where the adversary is additionally granted access to a verification oracle $\mathsf{Vrfy}(k, \cdot, \cdot)$.

(i) Make the above definition precise, using the formalism we used in class. Call the new notion "unforgeability under chosen message and verification attacks" (UF-CMVA).

(ii) Show that whenever a MAC has unique tags (i.e., for every key k there is only one valid tag ϕ for each message m) then UF-CMA implies UF-CMVA.

(iii) Show that if tags are not unique there exists a MAC that satisfies UF-CMA but not UF-CMVA.

(**Hint:** Given an arbitrary MAC $\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$ satisfying UF-CMA construct a contrived MAC $\Pi' = (\mathsf{Mac}', \mathsf{Vrfy}')$ with non-unique tags such that Π' is still UF-CMA but an attacker with access to a verification oracle can leak the entire secret key.)

(b) Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a PRF family. Analyze the following construction of a MAC with key space $\mathcal{K} = \{0,1\}^\lambda$ and message space $\mathcal{M} = \{0,1\}^{2n}$: $\mathsf{Mac}(k, m_1||m2) = F_k(m_1)||F_k(F_k(m_2))$, with $m_1, m_2 \in \{0,1\}^n$.

(c) Recall that in CBC-MAC the tag of a message $m = (m_1, \ldots, m_t) \in (\{0,1\}^n)^t$ is the value $\phi_t \in \{0,1\}^n$ computed using the following recursive equations: $\forall i \in [t], \phi_i = F_k(m_i \oplus \phi_{i-1})$ with $\phi_0 = 0^n$ and where F_k is sampled from a PRF family. Establish whether the following modifications of CBC-MAC are secure or not.

(i) Using CBC-MAC directly for authenticating variable-length messages.

(ii) A variant of CBC-MAC where, each time a tag is computed, a different value ϕ_0 is sampled uniformly at random from $\{0,1\}^n$ and output together with ϕ_t.

(iii) A variant of CBC-MAC where the output consists of all values $\phi_0, \phi_1, \ldots, \phi_t$.

3 Hashing — 20 Points

(a) Let $\mathcal{H} = \{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s \in \{0,1\}^\lambda}$ be a family of collision-resistant hash functions compressing $2n$ bits into n bits. Answer the following questions.

(i) Show that $\mathcal{H}$ is a seeded one-way function in the following sense: For all PPT adversaries $\mathcal{A}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\Pr\left[H_s(x') = y : \ s \leftarrow \{0,1\}^\lambda; x \leftarrow \{0,1\}^{2n}; y = H_s(x); x' \leftarrow \mathcal{A}(s, y)\right] \leq \nu(n).$$

(ii) What happens in case the set of functions $\mathcal{H}$ is not compressing (i.e., the domain of each function H_s is also $\{0,1\}^n$)? Does collision resistance imply one-wayness in this case?

(b) Let $\mathcal{H} = \{H_s : \{0,1\}^{4n} \to \{0,1\}^{2n}\}_{s\in\{0,1\}^\lambda}$ and $\mathcal{H}' = \{H'_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be families of collision-resistant hash functions. Analyse the following candidate hash function family compressing $4n$ bits into n bits: $\mathcal{H}^* := \{H^*_{s_1,s_2}\{0,1\}^{4n} \to \{0,1\}^n\}_{s_1,s_2\in\{0,1\}^\lambda}$ such that $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ for $s_1, s_2 \leftarrow\!\!\$\ \{0,1\}^\lambda$.

4 Number Theory — 20 Points

(a) Recall that the CDH problem asks to compute g^{ab} given (g, g^a, g^b) for $(\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$ and $a, b \leftarrow\!\!\$\ \mathbb{Z}_q$. Prove that the CDH problem is equivalent to the following problem: Given (g, g^a) compute g^{a^2}, where $(\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$ and $a \leftarrow\!\!\$\ \mathbb{Z}_q$.

(b) Let $N = p \cdot q$ be an RSA modulus. Show how one can find p and q given $(N, \varphi(N))$ without factoring N. (Recall that $\varphi(N) := (p-1)(q-1)$.)

Apply the above fact to the following setting: $N = 18830129$ and $\varphi(N) = 18819060$.

(c) Alice and Bob belong to the same organization and are given public keys e_A and e_B, respectively, corresponding to a common public RSA modulus N. Assume that e_A and e_B are relatively prime. Let $c_A = m^{e_A} \bmod N$ and $c_B = m^{e_B} \bmod N$ be two RSA encryptions of the same message $m \in \mathbb{Z}^*_N$, under public keys e_A and e_B (respectively). Prove that an eavesdropper given e_A, e_B, c_A and c_B can recover m.

(**Hint:** Use Bézout's identity.)

5 Public-Key Encryption — 20 Points

(a) Prove formally that CCA1 security[2] implies CPA security for any PKE scheme. On the other hand, show that there exists a PKE scheme that is CPA secure but not CCA1 secure.

(b) Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\{0,1\}$ (i.e., for encrypting a single bit). Consider the following natural construction of a multi-bit PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\{0,1\}^t$, for some polynomial $t = t(\lambda)$: (i) The key generation stays the same, i.e. $\mathsf{KGen}'(1^\lambda) = \mathsf{KGen}(1^\lambda)$; (ii) Upon input $m = (m[1], \ldots, m[t]) \in \{0,1\}^t$ the encryption algorithm $\mathsf{Enc}'(pk, m)$ outputs a

[2]Recall that in CCA1 security the adversary is given access to a decryption oracle prior to receiving the challenge ciphertext, whereas in CCA2 security the attacker is allowed to make decryption queries even after being given the challenge ciphertext (except that it cannot query the oracle on the challenge itself).

ciphertext $c = (c_1, \ldots, c_t)$ where $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk, m[i])$ for all $i \in [t]$; (iii) Upon input a ciphertext $c = (c_1, \ldots, c_t)$ the decryption algorithm $\mathsf{Dec}'(sk, c)$ outputs the same as $(\mathsf{Dec}(sk, c_1), \ldots, \mathsf{Dec}(sk, c_t))$.

(i) Show that if Π is CCA1 secure, so is Π'.

(ii) Show that, even if Π is CCA2 secure, Π' is not CCA2 secure.

(c) Recall the Padded RSA PKE scheme. Let $N = p \cdot q$, and e, d be such that $d \cdot e \equiv 1 \bmod (p-1)(q-1)$. The public key is $pk := (N, e)$ and the secret key is $sk := (N, d)$. Upon input a message $m \in \{0,1\}^{\ell(n)}$, pick a random $r \leftarrow \{0,1\}^{|N|-\ell(n)-1}$ and output $c := (r||m)^e \bmod N$. To decrypt a ciphertext $c \in \mathbb{Z}_N^*$ compute $\hat{m} := c^d \bmod N$, parse $\hat{m} := r||m$, and return m.

Show that Padded RSA is not CCA2 secure, by exhibiting a concrete chosen-ciphertext attack and analyzing its success probability.

(**Hint:** For simplicity, you may assume that the answer to a decryption query consists of the entire padded message $r||m$ and not just of the last ℓ significant bits of it. You will get a bonus if you can attack the scheme without making this assumption.)

Cryptography—Homework 1*

Sapienza University of Rome
Master Degree in Computer Science
Master Degree in Cybersecurity

Daniele Venturi

Due Date: November 15, 2017

1 Perfect Secrecy and One-Time Pad — 20 Points

(a) Note that whenever the all-zero key is chosen in the one-time pad, we obtain $\mathsf{Enc}(k, m) = 0^\ell \oplus m = m$. Is this a problem? In particular, suppose to modify the one-time pad by requiring that the key is sampled from the set $\mathcal{K}' := \{0,1\}^\ell \setminus \{0^\ell\}$. Is the resulting encryption scheme still perfectly secure? Prove your answer.

(b) Prove or refute: An encryption scheme with message space $\mathcal{M}$ is perfectly secret if and only if for all distributions M over $\mathcal{M}$, for all $m, m' \in \mathcal{M}$, and for every $c_0, c_1 \in \mathcal{C}$, we have:

$$\Pr[C = c_0] = \Pr[C = c_1].$$

(c) Define an appropriate notion of a 2-time ε-secure MAC, and give a construction that meets your definition.

2 Universal Hashing — 15 Points

(a) A family of functions $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ is called *universal* if for all distinct inputs $x, x' \in \mathcal{X}$ we have:

$$\Pr\left[h_s(x) = h_s(x') : \ s \leftarrow\!\!\$\ \mathcal{S}\right] = |\mathcal{Y}|^{-1}.$$

Show that any family $\mathcal{H}$ that is pairwise independent (as defined in class) is also universal.

*Some of the exercises are taken from the book *"Introduction to Modern Cryptography"* (second edition), by Jonathan Katz and Yehuda Lindell.

(b) Let $\ell, n > 0$. Consider the family of hash functions

$$\mathcal{H} = \{h_{\mathbf{A},\mathbf{b}} : \{0,1\}^n \to \{0,1\}^\ell\}_{\mathbf{A}\in\{0,1\}^{\ell\times n},\mathbf{b}\in\{0,1\}^\ell},$$

defined by $h_s(\mathbf{m}) = \mathbf{A}\cdot\mathbf{m} \oplus \mathbf{b}$, where $s = (\mathbf{A},\mathbf{b}) \in \{0,1\}^{\ell\times n} \times \{0,1\}^\ell$ and $\mathbf{m} \in \{0,1\}^n$, and where $\mathbf{A}, \mathbf{b}, \mathbf{m}$ are interpreted as matrices/vectors and all operations are performed modulo 2. Prove that $\mathcal{H}$ is pairwise independent.

(c) Let $\ell, n > 0$. Consider the family of hash functions $\mathcal{H}$ that is identical to the one in the above exercise, except that the matrix $\mathbf{A}$ is now sampled from the set $\mathbb{T}^{\ell\times n}$ of $\ell \times n$ Toeplitz matrices, i.e. all matrices $\mathbf{A} = (a_{i,j})$ such that $a_{i,j} = a_{i-1,j-1}$ when $i, j > 1$ (this means that the values along any diagonal are all equal).

Prove that $\mathcal{H}$ is still pairwise independent. What is the advantage w.r.t. the previous construction?

3 Negligible and Noticeable Functions — 15 Points

(a) Recall that a function $\nu : \mathbb{N} \to [0,1]$ is negligible if $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda) \in \text{poly}(\lambda)$. Show that the following alternative definition is equivalent: A function $\nu : \mathbb{N} \to [0,1]$ is negligible if for all $c \in \mathbb{N}$, there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda \geq \lambda_0$ we have $\nu(\lambda) < \lambda^{-c}$.

(b) Prove that $\nu(\lambda) = 2^{-\lambda}$ is negligible.

(c) A function $\mu : \mathbb{N} \to [0,1]$ is noticeable if there exists $c, \lambda_0 \in \mathbb{N}$ sucht that, for all $\lambda \geq \lambda_0$, we have $\mu(\lambda) \geq \lambda^{-c}$.

Explain the difference between a noticeable function and a non-negligible function. Show that the following function is both non-negligible and non-noticeable:

$$f(\lambda) = \begin{cases} 2^{-\lambda} & \text{if } \lambda \text{ is even} \\ \lambda^{-3} & \text{if } \lambda \text{ is odd.} \end{cases}$$

4 One-Way Functions — 25 Points

(a) Let $f : \{0,1\}^n \to \{0,1\}^n$ be a candidate OWF. Show that there always exists an inefficient attacker $\mathcal{A}_1$ inverting the function with probability one, and an efficient attacker $\mathcal{A}_2$ inverting the function with probability 2^{-n}.

(b) Analyze the following candidate OWFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $f(x, y) = x + y$, where $|x| = |y|$ and x, y are interpreted as natural numbers.

(ii) $g(x_1, x_2) = (f(x_1), x_2)$, where f is a OWF and $|x_1| = |x_2|$.

(iii) $g(x) = (f(x), f(f(x)))$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.

(iv) $g(x) = f(x||0)$, where $f : \{0,1\}^n \to \{0,1\}^n$ is a OWF.

5 Pseudorandom Generators — 20 Points

(a) Show that no PRG can be secure against computationally unbounded adversaries. In particular, let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling PRG. Show that there is an exponential-time distinguisher breaking the PRG with probability almost one.

(b) Analyze the following candidate PRGs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $G'(s) = G(s_1)|| \ldots ||G(s_n)$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and $s = s_1|| \ldots ||s_n \in \{0,1\}^{\lambda n}$.

(ii) $G'(s) = G(s||0^{|s|})$, where $G : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda+\ell}$ is a PRG.
(**Hint:** Consider the contrived PRG $G(s)$ that ignores the first half of its input, and returns $\hat{G}(s_{\lambda+1}, \ldots, s_{2\lambda})$ where $\hat{G}$ is itself a PRG stretching λ bits into $2\lambda + \ell$ bits. Argue that G is a PRG, but G' as defined in the exercise is not.)

6 Pseudorandom Functions — 10 Points

Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $F_k(x) = G'(k) \oplus x$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and G' denotes the output of G truncated to λ bits.

(ii) $F_k(x) := F_x(k)$, where $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a PRF.

(iii) $F'_k(x) = F_k(x||0)||F_k(x||1)$, where $x \in \{0,1\}^{n-1}$.

7 Secret-Key Encryption — 20 Points

Let $\mathcal{P} = \{P_k : \{0,1\}^{2n} \to \{0,1\}^{2n}\}_{k \in \{0,1\}^\lambda}$ be a family of *strong* PRPs. Show that the following construction of an SKE scheme $(\mathsf{Enc}, \mathsf{Dec})$ satisfies CCA security but it is not a secure authenticated encryption scheme.

Encryption: Upon input $m \in \{0,1\}^n$, sample $r \leftarrow\!\!\$\ \{0,1\}^n$ and return $c = \mathsf{Enc}(k, m) = P_k(m||r)$.

Decryption: Upon input $c \in \{0,1\}^{2n}$, let $m||r = P^{-1}(c)$ and output m.

8 Message Authentication — 25 Points

(a) Consider a variant of the game defining UF-CMA security of a MAC $\Pi = (\mathsf{Tag}, \mathsf{Vrfy})$, with key space $\mathcal{K} = \{0,1\}^\lambda$, where the adversary is additionally granted access to a verification oracle $\mathsf{Vrfy}(k, \cdot, \cdot)$.

(i) Make the above definition precise, using the formalism we used in class. Call the new notion "unforgeability under chosen message and verification attacks" (UF-CMVA).

(ii) Show that whenever a MAC has unique tags (i.e., for every key k there is only one valid tag ϕ for each message m) then UF-CMA implies UF-CMVA.

(iii) Show that if tags are not unique there exists a MAC that satisfies UF-CMA but not UF-CMVA.

(**Hint:** Given an arbitrary MAC $\Pi = (\mathsf{Tag}, \mathsf{Vrfy})$ satisfying UF-CMA construct a contrived MAC $\Pi' = (\mathsf{Tag}', \mathsf{Vrfy}')$ with non-unique tags such that Π' is still UF-CMA but an attacker with access to a verification oracle can leak the entire secret key.)

(b) Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a PRF family. Analyze the following construction of a MAC with key space $\mathcal{K} = \{0,1\}^\lambda$ and message space $\mathcal{M} = \{0,1\}^{2n}$: $\mathsf{Tag}(k, m_1||m2) = F_k(m_1)||F_k(F_k(m_2))$, with $m_1, m_2 \in \{0,1\}^n$.

(c) Recall that in CBC-MAC the tag of a message $m = (m_1, \ldots, m_t) \in (\{0,1\}^n)^t$ is the value $\phi_t \in \{0,1\}^n$ computed using the following recursive equations: $\forall i \in [t], \phi_i = F_k(m_i \oplus \phi_{i-1})$ with $\phi_0 = 0^n$ and where F_k is sampled from a PRF family. Establish whether the following modifications of CBC-MAC are secure or not.

(i) Using CBC-MAC directly for authenticating variable-length messages.

(ii) A variant of CBC-MAC where, each time a tag is computed, a different value ϕ_0 is sampled uniformly at random from $\{0,1\}^n$ and output together with ϕ_t.

(iii) A variant of CBC-MAC where the output consists of all values $\phi_0, \phi_1, \ldots, \phi_t$.

Cryptography—Homework 2

Sapienza University of Rome
Master Degree in Computer Science
Master Degree in Cybersecurity

Daniele Venturi

Due Date: December 21, 2017

1 Luby-Rackoff — 30 Points

Recall that the Feistel permutation $\Psi_f : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined by $\Psi_f(X) := (R, L \oplus f(R))$, where $X = L||R$, with both $L, R \in \{0,1\}^n$, and $f : \{0,1\}^n \to \{0,1\}^n$ is a function. Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^\lambda}$ be a PRF family. An r-round Feistel network $\Psi_{\mathcal{F}}[r]$ is the concatenation of r Feistel permutations, each using an independent PRF $F_{k_i} : \{0,1\}^n \to \{0,1\}^n$ from the family $\mathcal{F}$. More precisely, for every choice of the keys $k_1, \ldots, k_r \leftarrow\!\!\$\ \{0,1\}^n$, a permutation $\Psi_{F_{k_1},\ldots,F_{k_r}}$ in $\Psi_{\mathcal{F}}[r]$ is defined as follows:

$$\Psi_{F_{k_1},\ldots,F_{k_r}}(X) := \Psi_{F_{k_r}}(\cdots \Psi_{F_{k_2}}(\Psi_{F_{k_1}}(X))\cdots).$$

Answer the following questions:

(a) Show that a $\Psi_{\mathcal{F}}[1]$ is not a PRP family.

(b) Show that a $\Psi_{\mathcal{F}}[2]$ is not a PRP family.

(c) Show that a $\Psi_{\mathcal{F}}[3]$ is not a *strong* PRP family.[1]

2 Hashing — 25 Points

(a) Let $\mathcal{H} = \{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be a family of collision-resistant hash functions compressing $2n$ bits into n bits. Answer the following questions.

[1] A family of permutations $\mathcal{P} = \{P_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^\lambda}$ is called a strong PRP if for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that the following holds:

$$\left|\Pr\left[\mathcal{D}^{P(k,\cdot),P^{-1}(k,\cdot)}(1^\lambda) = 1 :\ k \leftarrow\!\!\$\ \{0,1\}^\lambda\right] - \Pr\left[\mathcal{D}^{R(\cdot),R^{-1}(\cdot)}(1^\lambda) = 1 :\ R \leftarrow\!\!\$\ \$_n\right]\right| \leq \nu(\lambda),$$

where $\$_n$ is the set of all possible permutations over $\{0,1\}^n$.

(i) Show that $\mathcal{H}$ is a seeded one-way function in the following sense: For all PPT adversaries $\mathcal{A}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\Pr\left[H_s(x') = y : \ s \leftarrow \{0,1\}^\lambda; x \leftarrow \{0,1\}^{2n}; y = H_s(x); x' \leftarrow \mathcal{A}(s,y)\right] \leq \nu(n).$$

(ii) What happens in case the set of functions $\mathcal{H}$ is not compressing (i.e., the domain of each function H_s is also $\{0,1\}^n$)? Does collision resistance imply one-wayness in this case?

(b) Let $\mathcal{H} = \{H_s : \{0,1\}^{4n} \to \{0,1\}^{2n}\}_{s\in\{0,1\}^\lambda}$ and $\mathcal{H}' = \{H'_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be families of collision-resistant hash functions. Analyse the following candidate hash function family compressing $4n$ bits into n bits: $\mathcal{H}^* := \{H^*_{s_1,s_2}\{0,1\}^{4n} \to \{0,1\}^n\}_{s_1,s_2\in\{0,1\}^\lambda}$ such that $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ for $s_1, s_2 \leftarrow\!\!\$\ \{0,1\}^\lambda$.

3 Number Theory — 25 Points

(a) Recall that the CDH problem asks to compute g^{ab} given (g, g^a, g^b) for $(\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$ and $a, b \leftarrow\!\!\$\ \mathbb{Z}_q$. Prove that the CDH problem is equivalent to the following problem: Given (g, g^a) compute g^{a^2}, where $(\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$ and $a \leftarrow\!\!\$\ \mathbb{Z}_q$.

(b) Let $N = p \cdot q$ be an RSA modulus. Show how one can find p and q given $(N, \varphi(N))$ without factoring N. (Recall that $\varphi(N) := (p-1)(q-1)$.)

Apply the above fact to the following setting: $N = 18830129$ and $\varphi(N) = 18819060$.

(c) Alice and Bob belong to the same organization and are given public keys e_A and e_B, respectively, corresponding to a common public RSA modulus N. Assume that e_A and e_B are relatively prime. Let $c_A = m^{e_A} \bmod N$ and $c_B = m^{e_B} \bmod N$ be two RSA encryptions of the same message $m \in \mathbb{Z}_N^*$, under public keys e_A and e_B (respectively). Prove that an eavesdropper given e_A, e_B, c_A and c_B can recover m.

(**Hint:** Use Bézout's identity.)

4 Public-Key Encryption — 30 Points

(a) Consider the following relaxation of CCA security for PKE schemes, so-called CCA1, where the adversary can only access the decryption oracle prior to receiving the challenge ciphertext (whereas in CCA security, also known as CCA2, the attacker is allowed to make decryption queries even after being given the challenge ciphertext). Give a formal definition of CCA1 security for PKE schemes.

(b) Prove formally that CCA1 security implies CPA security for any PKE scheme. On the other hand, show that there exists a PKE scheme that is CPA secure but not CCA1 secure.

(c) Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\{0,1\}$ (i.e., for encrypting a single bit). Consider the following natural construction of a multi-bit PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\{0,1\}^t$, for some polynomial $t = t(\lambda)$: (i) The key generation stays the same, i.e. $\mathsf{KGen}'(1^\lambda) = \mathsf{KGen}(1^\lambda)$; (ii) Upon input $m = (m[1], \ldots, m[t]) \in \{0,1\}^t$ the encryption algorithm $\mathsf{Enc}'(pk, m)$ outputs a ciphertext $c = (c_1, \ldots, c_t)$ where $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk, m[i])$ for all $i \in [t]$; (iii) Upon input a ciphertext $c = (c_1, \ldots, c_t)$ the decryption algorithm $\mathsf{Dec}'(sk, c)$ outputs the same as $(\mathsf{Dec}(sk, c_1), \ldots, \mathsf{Dec}(sk, c_t))$.

(i) Show that if Π is CCA1 secure, so is Π'.

(ii) Show that, even if Π is CCA2 secure, Π' is not CCA2 secure.

(d) Recall the Padded RSA PKE scheme. Let $N = p \cdot q$, and e, d be such that $d \cdot e \equiv 1 \bmod (p-1)(q-1)$. The public key is $pk := (N, e)$ and the secret key is $sk := (N, d)$. Upon input a message $m \in \{0,1\}^{\ell(\lambda)}$, pick a random $r \leftarrow \{0,1\}^{|N|-\ell(\lambda)-1}$ and output $c := (r||m)^e \bmod N$. To decrypt a ciphertext $c \in \mathbb{Z}_N^*$ compute $\hat{m} := c^d \bmod N$, parse $\hat{m} := r||m$, and return m.

Show that Padded RSA is not CCA2 secure, by exhibiting a concrete chosen-ciphertext attack and analyzing its success probability.

(**Hint:** For simplicity, you may assume that the answer to a decryption query consists of the entire padded message $r||m$ and not just of the last ℓ significant bits of it. You will get a bonus if you can attack the scheme without making this assumption.)

5 Signature Schemes — 20 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with message space $\mathcal{M} = \{0,1\}^\ell$, for some fixed $\ell \in \mathbb{N}$. Consider a family of hash functions $\mathcal{H} = \{h_s : \{0,1\}^* \to \{0,1\}^\ell\}_{s \in \{0,1\}^\lambda}$.

Define the following derived signature scheme $\Pi' = (\mathsf{KGen}', \mathsf{Sign}', \mathsf{Vrfy}')$: (i) Algorithm $\mathsf{KGen}'(1^\lambda)$ returns (pk', sk') such that $pk' = (pk, s)$ and $sk' = (sk, s)$, where $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$ and $s \leftarrow\!\!\$\ \{0,1\}^\lambda$; (ii) Algorithm $\mathsf{Sign}'(sk', m)$ takes a message $m \in \{0,1\}^*$ of arbitrary length, and outputs $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk, h_s(m))$; (iii) Algorithm $\mathsf{Vrfy}'(pk', m, \sigma)$ returns the same as $\mathsf{Vrfy}(pk, h_s(m), \sigma)$. Prove that if Π is UF-CMA and $\mathcal{H}$ is collision-resistant, then Π' is UF-CMA.

6 Identification Schemes 20 Points

Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a canonical ID scheme with challenge space $\mathcal{B}_\lambda$. For any polynomial $t = t(\lambda)$, consider the following t-fold parallel repetition of Π, denoted $\Pi^t := (\mathsf{Setup}, \mathsf{P}^t, \mathsf{V}^t)$: First $(pk, sk) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$ is run and then $\mathsf{P}^t(pk, sk)$ and $\mathsf{V}^t(pk)$ interact by simply running t independent executions of the original ID scheme between $\mathsf{P}(pk, sk)$ and $\mathsf{V}(pk)$ in parallel.

A little more formally, let $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ and $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2)$ be the algorithms of the underlying canonical ID scheme, where recall that $\mathsf{V}_1(1^\lambda)$ simply returns a random challenge $\beta \leftarrow\!\!\$\ \mathcal{B}_\lambda$. An interaction between P^t and V^t results in a transcript $\vec{\tau} \leftarrow\!\!\$\ (\mathsf{P}^t(pk, sk) \leftrightarrows \mathsf{V}^t(pk))$, where $\vec{\tau} := (\vec{\alpha}, \vec{\beta}, \vec{\gamma})$ is computed as follows:

1. P^t runs $\alpha_i \leftarrow\!\!\$\ \mathsf{P}_1(pk, sk)$ for all $i \in [t]$ and forwards $\vec{\alpha} = (\alpha_1, \ldots, \alpha_t)$ to V^t.
2. V^t samples $\beta_i \leftarrow\!\!\$\ \mathcal{B}_\lambda$ for all $i \in [t]$ and forwards $\vec{\beta} = (\beta_1, \ldots, \beta_t)$ to P^t.
3. P^t runs $\gamma_i \leftarrow\!\!\$\ \mathsf{P}_2(pk, sk, \alpha_i, \beta_i)$ for all $i \in [t]$, and forwards $\vec{\gamma} = (\gamma_1, \ldots, \gamma_t)$ to V^t.
4. V^t returns 1 if and only if $\mathsf{V}_2(pk, (\alpha_i, \beta_i, \gamma_i)) = 1$ for all $i \in [t]$.

Answer the following questions.

(a) Show that Π^t is a canonical ID scheme. What is the challenge space?

(b) Prove that as long as Π satisfies completeness, special soundness, and honest-verifier zero knowledge, so does Π^t (for any polynomial $t(\lambda)$).

HOMEWORK 1 CRYPTOGRAPHY ALESSIA MARROCCO 1774884

EXERCISE 1

a) Recall to 3rd definition of perfect secrecy states that an encryption scheme is perfectly secret if the cyphertext distribution does not depend on the message. In other words, every message induces the same ciphertest distribution. The condition stated in the exercise implies that every message induces the uniform ciphertext distribution. Clearly, if that is true, then the scheme is also perfecly secret, since every messages induces the same ciphertext distribution (the uniform distribution). However, the condition is strictly stronger than perfect secrecy, since there are perfectly secret schemes that do not meet the condition. Sinee the question was if the condition is equivalent to perfect secrecy (i.e. "if and only if"), the statement is wrong.

For example, consider modifying the one-time pad so encryption appends a bit that is 0 with probability $1/4$ and 1 with probability $3/4$. This scheme will still be perfectly secret, but ciphertexts ending in 1 are more likely than ciphertexts ending in 0.

b) $\Pi = (Enc, Gen)$ respects PS in $\mathcal{M}, K$

$\exists t \in \mathbb{N} : \forall m \in \mathcal{M} \Rightarrow Pr[Dec(K, Enc(K,m)) = m] \geq 2^{-t}$

Prove $|K| \geq |\mathcal{M}| \cdot 2^{-t}$

Proof. Perfect Secrecy implies that $|\mathcal{M}|^{-1} = Pr[M = m \mid C = c]$

Continuing we have

$$\begin{aligned} Pr[M = m \mid C = c] &\geq Pr[Enc(K,m) = c] \\ &= Pr[K = k \mid C = c] \cdot Pr[Dec(K, Enc(K,m)) = m] \\ &\geq |K|^{-1} \cdot Pr[Dec(K, Enc(K,m)) = m] \\ &\geq |K|^{-1} \cdot 2^{-t}, \end{aligned}$$

by the correctness guarantee, and using the fact that m and K are indipendent. We conclude that $|K| \geq |\mathcal{M}| \cdot 2^{-t}$

EXERCISE 2

a) A family $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in S}$ of hash functions is called t-wise indipendent if for all sequences of distinct input $x_1, \dots, x_t \in \mathcal{X}$, and for any output sequence $y_1, \dots y_t \in \mathcal{Y}$ (not necessarily distinct), we have that:

$$Pr[h_s(x_1) = y_1 \wedge \dots \wedge h_s(x_t) = y_t : s \leftarrow\!\!\$\; S] = \frac{1}{|\mathcal{Y}|^t}$$

i. for any $t \geq 2$, show that if $\mathcal{H}$ is t-wise indipendent, then it is also $(t-1)$-wise indipendent

Proof. Fix a sequence of distinct $x_1, \dots, x_{t-1} \in X$ and a sequence $y_1, \dots, y_{t-1} \in Y$. Fix $x \in X \setminus \{x_1, \dots, x_{t-1}\}$ and write

$$Pr[h_s(x_1) = y_1 \wedge \dots \wedge h_s(x_{t-1}) = y_{t-1}] = \sum_{y \in Y} Pr[h_s(x_1) = y_1 \wedge \dots \wedge h_s(x_{t-1}) = y_{t-1} \wedge h_s(x) = y] =$$

$$= \sum_{y \in Y} \frac{1}{|\mathcal{Y}|^t} = \frac{|\mathcal{Y}|}{|\mathcal{Y}|^t} = \frac{1}{|\mathcal{Y}|^{t-1}}$$ then the propositions holds because $x \in X \setminus \{x_1, \dots, x_{t-1}\}$

ii. Let q be a prime. Show that the family $\mathcal{H} = \{h_s : \mathbb{Z}_q \to \mathbb{Z}_q\}_{s \in \mathbb{Z}_q^3}$, defined by $h_s(x) = h_{s_0,s_1,s_2}(x) = s_0 + s_1 x + s_2 x^2 \bmod q$, is 3-wise indipendent.

To prove this we need a theorem and a definition.

DEF. A square Vandermonde matrix is a matrix in the form

$$V_n = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix}$$

THM. Let be V_n a square Vandermonde matrix, if the $x_0, x_1, \dots, x_{n-1}$ are all distinct, then V_n is invertible.

Let's prove this for $n=3$

$$V_3 = \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} \Rightarrow \det V_3 = \begin{vmatrix} 1 & x_0 & x_0^2 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 \\ 0 & x_2 - x_0 & x_2^2 - x_0^2 \end{vmatrix} =$$

$$= (x_1 - x_0)(x_2 - x_0)(x_2 + x_0) - (x_2 - x_0)(x_1 - x_0)(x_1 + x_0) =$$

$$= (x_1 - x_0)(x_2 - x_0)(x_2 + x_0 - x_1 - x_0) = (x_1 - x_0)(x_2 - x_0)(x_2 - x_1)$$

Finally if x_0, x_1, x_2 are all distinct, $\det(V_3) \neq 0$ and V_3 is invertible.

So given $h_s \in \mathcal{H}_n$, $s = (s_0, \dots, s_{n-1})$ we can evaluate h_s in $x_0, \dots, x_{n-1} \in \mathbb{Z}_q$ simultaneously by constructing the matrix V_n and calculating the product $y = V_n \cdot s$

If $x_0, \dots, x_{n-1} \in \mathbb{Z}_q$ are all distinct V_n is invertible by theorem, thus we can write $s = V_n^{-1} y$. So we can rewrite the event $h_s(x_0) = y_0 \wedge \cdots \wedge h_s(x_{n-1}) = y_{n-1}$ as $V_n s = y$

Finally being s uniform over $\mathbb{Z}_q^n$ and being every y mapped into exactly one $z = V_n^{-1} y \in \mathbb{Z}_q^n$ and vice versa, we have that

$$\Pr[h_s(x_0) = y_0 \wedge \cdots \wedge h_s(x_{n-1}) = y_{n-1}] = \Pr[s = V_n^{-1} y] = \Pr[s = z] = \frac{1}{|\mathbb{Z}_q|^n}$$

So to prove that for $q \geq 3$, q prime, $\mathcal{H}_3 = \{h_s : \mathbb{Z}_q \to \mathbb{Z}_q : x \mapsto s_0 + s_1 x + s_2 x^2\}_{s \in \mathbb{Z}_q^3}$ is 3-wise indipendent, we can consider it is a particular case with $n = 3$.

b) X is a (K, n)-source if $X \in \{0,1\}^n$, and the min-entrophy of X is at least K.

i. suppose that $\ell = 128$, the minimal amount of min-entropy needed in order to obtain $\varepsilon = 2^{-80}$, applying the leftover hash lemma is

$$K \geq \ell + 2 \cdot \log_2 \frac{1}{\varepsilon} - 2 = 128 + 2 \cdot \log_2 2^{80} - 2 = 128 + 160 - 2 = 286 \Rightarrow K \geq 286$$

the entropy loss δ is $\delta = 2 \cdot \log_2 \frac{1}{\varepsilon} - 2 = 2 \cdot \log_2 2^{80} - 2 = 2 \cdot 80 - 2 = 158$

ii. suppose that $K = 238$, the maximal amount of uniform randomness that you can obtain with $\varepsilon = 2^{-80}$, applying the leftover hash lemma is

$$K \geq \ell + 2 \cdot \log_2 \varepsilon^{-1} - 2 \Rightarrow \ell \leq K - 2 \log_2 \varepsilon^{-1} + 2 = 238 - 2 \cdot 80 + 2 = 80 \Rightarrow \ell \leq 80$$

EXERCISE 3

a) Let $G: \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a PRG with λ-bit stretch. Prove that G is by itself a one-way function.

Proof. Assume that there exists a PPT adversary A which breaks the one-wayness property of G. We can use this A to create an efficient attack A' against the pseudorandomness of G. Let algorithm A', on input $y \in \{0,1\}^{2\lambda}$, get $z = A(y)$ by running A on y, and test if $G(z) = y$. If $G(z) = y$, A' decides that y is pseudorandom and outputs 1. Otherwise A' decides that y is truly random and outputs 0. By assumption on A, the probability

$$Pr[G(z) = y \mid x \leftarrow \{0,1\}^\lambda ; y = G(x) ; z = A(y)] \geq \varepsilon(\lambda)$$

where $\varepsilon(\lambda)$ is some non-negligible function of λ. By the above description of A' in terms of A we have:

$$Pr[A'(y) = 1 \mid x \leftarrow \{0,1\}^\lambda ; y = G(x)] = Pr[G(z) = y \mid x \leftarrow \{0,1\}^\lambda ; y = G(x) ; z = A(y)]$$

And therefore

$$Pr[A'(y) = 1 \mid x \leftarrow \{0,1\}^\lambda ; y = G(x)] \geq \varepsilon(\lambda) \quad (4)$$

So, on pseudorandom y's, A' answer 1 with probability at least $\varepsilon(\lambda)$.

$$Pr[A'(y) = 1 \mid y \leftarrow \{0,1\}^{2\lambda}] \leq \frac{1}{2^\lambda} \quad (5)$$

because A' answer 1 only if A(y) return z s.t. $G(z) = y$. But note that the range of $G: \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ can only have 2^λ elements, and therefore if you pick random $y \leftarrow \{0,1\}^{2\lambda}$, the probability that there exists $z \in \{0,1\}^\lambda$ s.t. $G(z) = y$ is at most $2^\lambda / 2^{2\lambda} = 1/2^\lambda$.

Taking together equations (4) and (5) we get that G must be an insecure PRG, because we have an efficient A' s.t.

$$Pr[A'(y) = 1 \mid x \leftarrow \{0,1\}^\lambda ; G(x) = y] - Pr[A'(y) = 1 \mid y \leftarrow \{0,1\}^{2\lambda}] \geq \varepsilon(\lambda) - \frac{1}{2^\lambda}$$

which is non-negligible. Since this contradicts our assumptions if follows that G must indeed by a OWF.

b) Let $f: \{0,1\} \to \{0,1\}$ be a OWF. Consider the function $g: \{0,1\}^{n+\log n} \to \{0,1\}^{n+\log n+1}$ defined by $g(x \| j) := (f(x), j, x_j)$, where $x := (x_1, \dots, x_n)$ and j is interpreted as an integer in $[n]$ (i.e. $|j| = \log n$)

i. Show that g is a OWF if f is

ii. Show that for every $i \in [n']$ there is a PPT algorithm A_i for which

$$Pr[A_i(g(x')) = x_i' : x' \leftarrow \{0,1\}^{n+\log n}] \geq \frac{1}{2} + \frac{1}{2n} \quad \text{where } x' = (x_1', \dots, x_{n'}')$$

Proof. Given $g(x)$, if a PPT A could invert g finding $(x', \langle j' \rangle)$ such that $g(x, \langle j \rangle) = g(x', \langle j' \rangle)$, A would have found in particular x' such that $f(x') = f(x)$ inverting the OWF f.

Now that we constructed the OWF g for all $i \in \{1, \dots, n+\log n\}$ we can construct A_i algorithm as follows

1. A_i gets the challenge to find $x' = x \| \langle j' \rangle$ given $g(x')$
2. if $i > n$, A_i outputs $(g(x'))_i = j_{i-n} = x_i'$
3. if $i \leq n$, but $i = j$ A_i outputs $(g(x'))_{n+\log n+1} = x_j = x_i$
4. if $i \leq n$ and $i \neq j$ A_i outputs $b \leftarrow \$ \{0,1\}$

Thus for $i > m$, the probability to win for A_i is exactly 1, on the other hand, for $i \leq m$ we have

$$Pr[A_i(g(x\|\langle j\rangle)) = x_i'] = Pr[A_i(g(x\|\langle j\rangle)) = x_i' \wedge i = j] + Pr[A_i(g(x\|\langle j\rangle)) = x_i' \wedge j \neq j] =$$

$$= Pr[A_i(g(x\|\langle i\rangle)) = x_i'] Pr[j = i] + Pr[A_i(g(x\|\langle j\rangle)) = x_i' \mid i \neq j] \cdot Pr[j \neq i] =$$

$$= 1 \cdot \frac{1}{2^{\log m}} + \frac{1}{2} \frac{2^{\log m} - 1}{2^{\log m}} = \frac{1}{n} + \frac{1}{2} - \frac{1}{2n} = \frac{1}{2} + \frac{1}{2n}$$

Putting the two together, for all $i \in \{1, \ldots, n + \log n\}$ it is possible to construct a PPT algorithm A_i such that

$$Pr[A_i(g(x\|\langle j\rangle)) = x_i'] \geq \frac{1}{2} + \frac{1}{2n}$$

thus breaking the randomicity of g (i.e. it is not possibile to claim that every OWF hides at least one specific bit of the input).

EXERCISE 4

a) Let $G_1, G_2 : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$, $\lambda \geq 1$. At least one of them is a secure PRG. Show how to design a secure PRG $G^* : \{0,1\}^{2\lambda} \to \{0,1\}^{\lambda+\ell}$ by combining G_1 and G_2

We can construct it by putting $G^*(x_1, x_2) = G_1(x_1) \oplus G_2(x_2)$.

In fact, if G_1 is a PRG, then

$$G^*(U_\lambda, U'_\lambda) = G_1(U_\lambda) \oplus G_2(U'_\lambda) \approx_c U_{\lambda+\ell} \oplus Y \approx_c U_{\lambda+\ell}$$

where $Y = G_2(U'_\lambda)$. Otherwise, G_2 must be a PRG and

$$G^*(U_\lambda, U'_\lambda) = G_1(U_\lambda) \oplus G_2(U'_\lambda) \approx_c X \oplus U_{\lambda+\ell} \approx_c U_{\lambda+\ell}$$

where $X = G_1(U_\lambda)$.

b) Can you prove that your construction works when using the same seed $s^* \in \{0,1\}^\lambda$ for both G_1 and G_2.

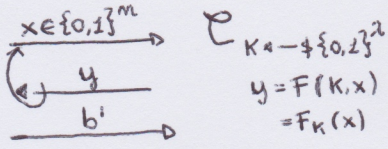
If $G_1 = G_2$ (or more in general, G_1 and G_2 have some bits in common), we get $G^*(x) = G_1(x) \oplus G_2(x) = 0$ (or, more in general, the bits that G_1 and G_2 have in common are 0 after the XOR).

EXERCISE 5

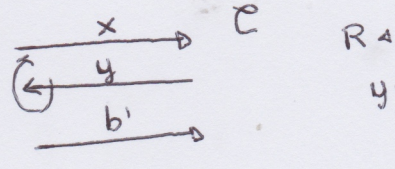
a) Let $\mathcal{F}$ be a PRF family and consider a computionally unbounded distinguisher $\mathcal{D}$. In particular $\mathcal{D}$ knows $f_K(x)$ for every $x \in \{0,1\}^n$, for every $K \in \{0,1\}^\lambda$ and for every $f_K \in \mathcal{F}$.

The idea is that with polynomial number of queries, $\mathcal{D}$ can restrict the domain and try to guess K if $\mathcal{C}$ is playing the $REAL_{\mathcal{F},\mathcal{D}}$ game, thus answering REAL; otherwise, $\mathcal{D}$ answers RAND.

$GAME^{real}_{\mathcal{F},A}(\lambda)$



$GAME^{rand}_{\mathcal{F},A}(\lambda)$



Thm We have this definition: Family $\mathcal{F}$ is a PRF is $GAME^{rand}_{\mathcal{F},A}(\lambda) \approx_c GAME^{real}_{\mathcal{F},A}(\lambda)$

b) i. $F_K(x) = G'(K) \oplus x$, where $G: \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and G' denotes the output of G truncated λ bits

This candidate does not yield a PRF. Let $\mathcal{D}$ be a distinguisher that is given oracle access to either $F_K(\cdot)$ for a random $K \in \{0,1\}^\lambda$, or to a truly random function $R: \{0,1\}^\lambda \to \{0,1\}^\lambda$. The $\mathcal{D}$ queries its oracle upon input any two distincts values $x, x' \in \{0,1\}^\lambda$, receiving back outputs y, y'. Hence, $\mathcal{D}$ returns 1 if and only if $x \oplus x' = y \oplus y'$.

Clearly, if $\mathcal{D}$'s oracle is $F_K(\cdot)$ (for a uniform $K \leftarrow\!\!\$ \{0,1\}^\lambda$):

$$y \oplus y' = (G'(K) \oplus x) \oplus (G'(K) \oplus x') = x \oplus x'$$

and thus $\mathcal{D}$ always outputs 1 in such a case. On the other hand, if $\mathcal{D}$'s oracle is R, the probability that $x \oplus x' = R(x) \oplus R(x')$ is equal to $2^{-\lambda}$. Hence, $\mathcal{D}$ has distinguish advantage nearly equal to 1.

ii. $F_K(x) := F_x(K)$, where $F: \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a PRF.

This candidate does not yield a PRF. Consider the contrived PRF F that upon input the all-zero key 0^λ always returns 0^ℓ indipendently of the input, i.e. $F_{0^\lambda}(x) = 0^\ell$ for all $x \in \{0,1\}^\lambda$. On the one hand, it is easy to prove that the above function still defines a PRF. On the other hand, it is easy to distinguish $F_x(K)$ (for $K \leftarrow\!\!\$ \{0,1\}^\lambda$) from a truly random function by simply querying input $x = 0^\lambda$ to the target oracle: if the oracle implements the PRF, we will obtain 0^ℓ with probability 1, whereas if the oracle implements a truly random, function $R: \{0,1\}^\lambda \to \{0,1\}^\ell$ the value 0^ℓ will be obtained with probability $2^{-\ell}$.

iii. $F'_K(x) = F_K(x\|0) \| F_K(x\|1)$, where $x \in \{0,1\}^{n-1}$

This candidate does yield a PRF. By contraddiction, assume there exists $\mathcal{D}'$ that can distinguish (oracle access to) $F'_K(\cdot)$ with a random key from a truly random function $R': \{0,1\}^{n-1} \to \{0,1\}^{n-1}$ with non negligible probability.

Consider the following distinguisher $\mathcal{D}$, whose goal is to distinguish $F_K(\cdot)$ with a random key from a truly random function $R: \{0,1\}^n \to \{0,1\}^n$: (1) upon input an oracle query $x' \in \{0,1\}^{n-1}$ from $\mathcal{D}$, define $x^0 := x' \| 0$ and $x^1 := x' \| 1$, query x^0, x^1 to the target oracle receiving back values y^0, y^1, and return $y^0 \| y^1$ to $\mathcal{D}'$; (2) Output the same as $\mathcal{D}'$. Clearly, $\mathcal{D}$ is roughly as efficient as $\mathcal{D}'$ (in fact, if $\mathcal{D}'$ makes q' queries, $\mathcal{D}$ needs to make $q = 2q'$), moreover it perfectly simulates the view of $\mathcal{D}'$ and thus it retains the same (non negligible) advantages.

EXERCISE 6

a) By definition of CPA-security, we can say that the encryption scheme $\Pi = (Enc, Dec)$, given t (time), q tries, A attacker and ε, it holds that

$$Pr\left[Exp^{SKE,\Pi}_{\text{experiment}}(A,q) = m\right] \leq \frac{1}{2} + \varepsilon$$

Which means that in a limited time and tries the probability is not $\varnothing$. If we have an adversary which is computionally unbounded we have

$$\sum_{t,q} P\left[Exp_{\Pi, SKE}(A,q) = m\right] = 1$$

EXERCISE 7

a) DEFINITION OF SUF-CMA

Let $MA = (K, T, V)$ be a message authentication scheme and A an adversary, set up a game $SUFCMA_{MA}$

- $K \leftarrow\$ K$, $S \leftarrow\$ \emptyset$ (inizialize)
- $T \leftarrow\$ T_K(M)$, $S \leftarrow S \cup \{(M,T)\}$ → returns T (Tag)
- $d \leftarrow V_K(M,T)$, if $(d = 1 \wedge (M,T) \notin S)$ → returns d (Verify)
- returns win (finalize)

suf-cma advantage is $Adv^{suf\text{-}cma}_{MA}(A) = P\left[SUFCMA^{A}_{MA} \Rightarrow true\right]$

Any MA schema $MA = (K, T, V)$ that is SUF-CMA schema is a sub-domain of UF-CMA.

Suppose A's Tag starts to query $M_1, \dots, M_q$, receiving

$T_1 \leftarrow\$ T_K(M_1)$, $T_2 \leftarrow\$ T_K(M_2)$, $\dots$, $T_q \leftarrow\$ T_K(M_q)$

(other notation) $T_1 \leftarrow\$ T(m_1, r_K) \dots T_q \leftarrow\$ T(m_q, r_K)$

And suppose A queries Vrfy (M,T). Then

$M \notin \{M_1, \dots, M_q\} \Rightarrow (M,T) \notin \{(m_1, r_1), \dots, (m_q, r_q)\}$

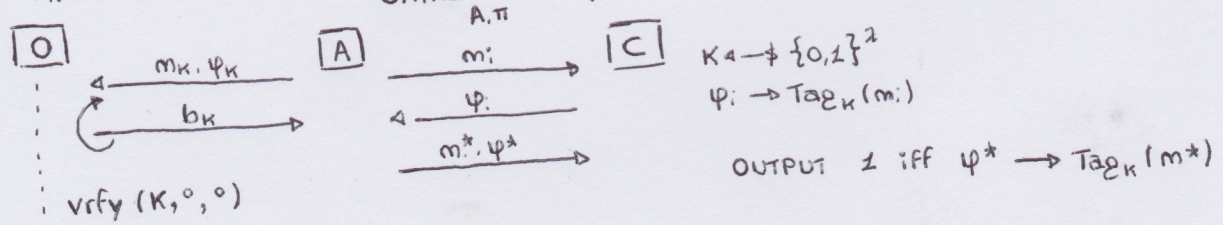
So, if A wins in game $UFCMA_{MA}$ it also wins in game $SUFCMA_{MA}$ but not vice-versa:

$$Adv^{ufcma}_{MA}(A) \leq Adv^{suf\text{-}cma}_{MA}(A)$$

b) i. let assume UF-CMVA security is associate with the following same scheme

$G^{UF\text{-}CMVA}_{\pi,A}(\lambda)$: 1) $K \leftarrow\$ \{0,1\}^\lambda$; 2) $(m^*, \varphi^*) \leftarrow A^{Tag_K(), Vrfy_K()}(1^\lambda)$

3) Output 1 if and only if $Vrfy_K(m^*, \varphi^*) = Accepted$ and m^* is fresh (never queried from A). We are assuming that A can make Q_T queries to $Tag_K()$ and Q_u queries to $Vrfy_K()$.

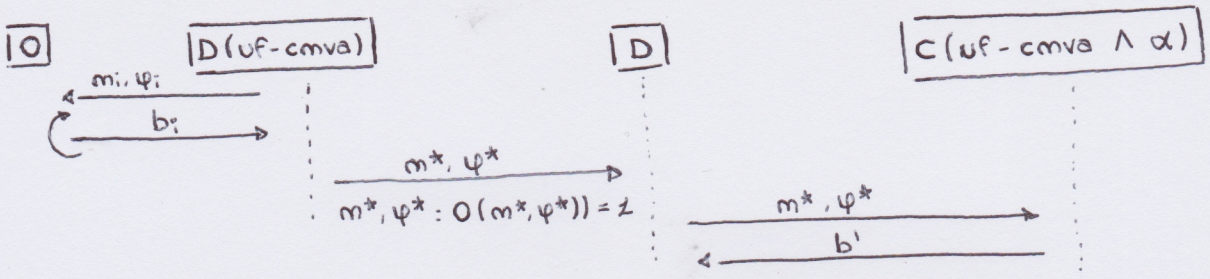


ii. let assume uf-cmva security is associated with the following game scheme and with the following property of "unique" which can call α

α $\forall K$, $\forall (m, m') \in M$ with $m \neq m'$: $Tag_K(m) = Tag_K(m')$ or $Tag_K(m) \neq Tag_K(m')$

$\forall K$, $\forall m \in M$, $\forall (K, m, \varphi)$, $\forall (K, m, \varphi')$ → $\varphi = \varphi'$

We have to show that: (uf-cma ∧ α) → uf-cmva

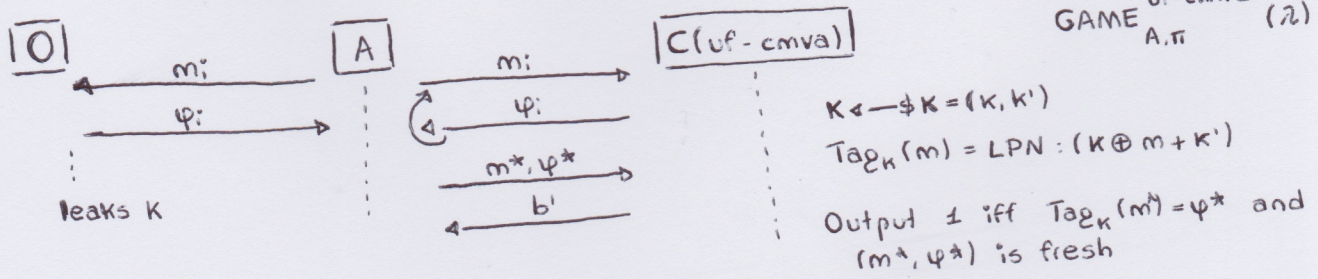


The oracle, for construction makes operation in poly(λ), so D(uf-cmva) it works also in poly(λ) and so forth D will be in poly(λ) so implies UF-CMVA.
Why? Because if Tag is probabilistic it can assume different values with high probability, and so the adversary cannot verify in poly time if φ is a possible Tag for the message.

iii. Let assume uf-cmva security is associate with the following game scheme and with the following property of "unique" which can call α

α $\forall K, \forall (m, m') \in M$, with $m \neq m'$: $Tag_K(m) = Tag_K(m')$ or $Tag_K(m) \neq Tag_K(m')$
$\forall K, \forall m \in M, \forall (K, m, \varphi), \forall (K, m, \varphi') \rightarrow \varphi = \varphi'$

We have to show that : (uf-cmva $\wedge$ not α) $\rightarrow$ not uf-cmva (but only uf-cma)



The general idea is that if the tag for each m is not unique, the probability of adversary to guess a message from a given φ is higher and so the verified attacks can not be guaranteed.

(1a) $\pi = (Enc, Dec)$

$\forall k \in K, \forall m \in M, \forall c_0, c_1 \in C \Rightarrow P[C=c_0] = P[C=c_1]$

Demonstration:

> 3rd th. of PS:
> $\forall m, m' \in M, \forall c \in C \Rightarrow P[Enc(K,m)=c] = P[Enc(K,m')=c]$

Knowing that M (Random Variable) and C (obtained with K, which is uniform) are independent, so we can say:

$P[M=m]\,P[C=c] = P[M=m \wedge C=c]$, we can imply that:

- $P[Enc(K,m)=c_0] = P[Enc(K,m)=\epsilon \mid C=c_0] = P[\epsilon = C]$
- $P[Enc(K,m)=c_1] = P[Enc(K,m)=\epsilon \mid C=c_1] = P[\epsilon = C]$

Which demonstrate the PS, but it wasn't the only way.

Refuted

(1b) $\pi = (Enc, Dec)$ respects PS in M, K

$\exists t \in N \mid \forall m \in M \Rightarrow P[Dec(K, Enc(K,m)) = m] \geq 2^{-t}$

$P[Dec(\mathbb{K}, Enc(\mathbb{K}, m)) = m \mid \mathbb{K} = K] \geq 2^{-t}$,

$P[Dec(\mathbb{K}, Enc(\mathbb{K}, m)) = m]\,P[\mathbb{K} = K] \geq 2^{-t}$

$P[Dec(\mathbb{K}, Enc(\mathbb{K}, M)) = M \mid M = m] \geq 2^{-t}\,|K|^{-1}$

$$\frac{P[M=m \,/\, Dec\ldots]^{0\text{ ou }1} \cdot P[\not{Dec}\ldots]^{0\text{ ou }1}}{P[M=m]_{|M|^{-1}}} \geq 2^{-t}\,|K|^{-1}$$

$\Rightarrow |K| \geq |M|\,2^{-t}$

2a-i

H is t-wise independent by definition. If we get (t-1) we have:

$$P_{1\dots(t-1)}\left[h_S(x_1)=y_1 \wedge \dots \wedge h_S(x_{t-1})=y_{t-1} : S \leftarrow\$ S\right] = \frac{1}{|y|^{t-1}}$$

if we take $m = t-1$, the property of t-wise independence is respected.

$$P_{1\dots m}\left[h_S(x_1)=y_1 \wedge \dots \wedge h_S(x_m)=y_m : S \leftarrow\$ S\right] = \frac{1}{|y|^{m}}$$

It is true for t, so it is true for t-1 too. (Even if $t-1=1$)

2a-ii

$h_S(x) := h_{s_0,s_1,s_2}(x) := s_0 + s_1 x + s_2 x^2 \bmod q$

$\forall x, x_1, y, y_1$ we want to find

$$P\left[s_2 x^2 + s_1 x + s_0 = y \wedge s_2 x_1^2 + s_1 x_1 + s_0 = y_1\right] \qquad (s_0, s_1, s_2 \leftarrow\$ \mathbb{Z}_p \text{ by definition})$$

$$\Rightarrow P\left[\begin{pmatrix} x^2 & x & 1 \\ x_1^2 & x_1 & 1 \end{pmatrix}\begin{pmatrix} s_2 \\ s_1 \\ s_0 \end{pmatrix} = \begin{pmatrix} y \\ y_1 \end{pmatrix}\right] =$$

$$\Rightarrow = P\left[\begin{pmatrix} s_2 \\ s_1 \\ s_0 \end{pmatrix} = \begin{pmatrix} x_2^2 & x & 1 \\ x_1^2 & x_1 & 1 \end{pmatrix}\begin{pmatrix} y \\ y_1 \end{pmatrix}\right] = \frac{1}{|y|^3}$$

True since $\begin{pmatrix} x_2 & x & 1 \\ x_1^2 & x_1 & 1 \end{pmatrix}\begin{pmatrix} y \\ y_1 \end{pmatrix}$ are constants, which should be verified by the (2a-i)

Cryptography – Homework 1

1 Perfect Secrecy

We have the following two definitions.

Definition 1.1 (Shannon)
A SKE scheme $\Pi = (\text{Enc}, \text{Dec})$ has (Shannon) **perfect secrecy** if

$$\forall m \in \mathcal{M}, \forall c \in \mathcal{C}, \quad \mathbf{Pr}\left[M = m \mid C = c\right] = \mathbf{Pr}\left[M = m\right].$$

Definition 1.2 (Eav)
Let $\Pi = (\text{Enc}, \text{Dec})$ be a SKE scheme and let $\textsc{Game}^{\text{eav}}_{\Pi,\mathscr{A}} : \mathscr{A} \longleftrightarrow \mathscr{C}$ be defined as follows:

1. the challenger $\mathscr{C}$ picks $k \leftarrow \$\mathcal{K}$ and $b \leftarrow \$\{0,1\}$;
2. the adversary $\mathscr{A}$ chooses $(m_0, m_1) \in \mathcal{M} \times \mathcal{M}$;
3. $\mathscr{C}$ computes $c = \text{Enc}(k, m_b)$ and gives c to $\mathscr{A}$;
4. $\mathscr{A}$ chooses a bit b';
5. the game outputs 1 if and only if $b = b'$.

We say that Π has (EAV) **perfect secrecy** if

$$\mathbf{Pr}\left[\textsc{Game}^{\text{eav}}_{\Pi,\mathscr{A}} = 1\right] = \frac{1}{2}.$$

Applying Shannon's Theorem, the Definition 1.1 is equivalent to require that

$$\forall m_0, m_1 \in \mathcal{M}, \forall c \in \mathcal{C}, \quad \mathbf{Pr}\left[\text{Enc}(K, m_0) = c\right] = \mathbf{Pr}\left[\text{Enc}(K, m_1) = c\right].$$

Claim 1.3
Definition 1.1 $\Rightarrow$ Definition 1.2.

Proof
Let B be the (uniformly distributed) random variable associated to $b \leftarrow \$\{0,1\}$ and let B' be the random variable associated to the choice of b' from the adversary $\mathscr{A}$. Then, proving the formula in the Definition 1.2 is equivalent to prove that

$$\frac{1}{2} = \mathbf{Pr}\left[B' = B\right],$$

where K is the (uniformly distributed) random variable associated to $k \leftarrow \$\mathcal{K}$.
Splitting in the two cases $B = 0$ and $B = 1$ and using the Bayes formula, we obtain

$$\begin{aligned}
\mathbf{Pr}\left[B' = B\right] &= \mathbf{Pr}\left[\text{Enc}(K, m_{B'}) = \text{Enc}(K, m_B)\right] \\
&= \mathbf{Pr}\left[\text{Enc}(K, m_{B'}) = \text{Enc}(K, m_B) \wedge B = 0\right] \\
&\quad + \mathbf{Pr}\left[\text{Enc}(K, m_{B'1}) = \text{Enc}(K, m_B) \wedge B = 1\right] \\
&= \mathbf{Pr}\left[\text{Enc}(K, m_{B'}) = \text{Enc}(K, m_B) \mid B = 0\right] \mathbf{Pr}\left[B = 0\right] \\
&\quad + \mathbf{Pr}\left[\text{Enc}(K, m_{B'}) = \text{Enc}(K, m_B) \mid B = 1\right] \mathbf{Pr}\left[B = 1\right].
\end{aligned}$$

Now remember that B is uniformly distributed over $\{0,1\}$, thus

$$\mathbf{Pr}\left[B=0\right]=\mathbf{Pr}\left[B=1\right]=\frac{1}{2},$$

and the above formula becames

$$\mathbf{Pr}\left[B'=B\right]=\frac{1}{2}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_0)\right]+\frac{1}{2}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_1)\right].$$

Call

$$C_0=\mathrm{Enc}(K,m_0),\qquad C_1=\mathrm{Enc}(K,m_1)$$

and split the two probabilities over all the possible choiches of $c\in\mathcal{C}$: we obtain

$$\begin{aligned}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_0)\right]&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=C_0\wedge C_0=c\right]\\&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=C_0\mid C_0=c\right]\mathbf{Pr}\left[C_0=c\right]\\&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=c\right]\mathbf{Pr}\left[C_0=c\right]\end{aligned}$$

and

$$\begin{aligned}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_1)\right]&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=C_1\wedge C_1=c\right]\\&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=C_1\mid C_1=c\right]\mathbf{Pr}\left[C_1=c\right]\\&=\sum_{c\in\mathcal{C}}\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=c\right]\mathbf{Pr}\left[C_1=c\right].\end{aligned}$$

Apply now Definition 1.1:

$$\mathbf{Pr}\left[C_0=c\right]=\mathbf{Pr}\left[C_1=c\right].$$

We obtain that the two above probabilities are equal:

$$\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_0)\right]=\mathbf{Pr}\left[\mathrm{Enc}(K,m_{B'})=\mathrm{Enc}(K,m_1)\right],$$

or, in other words,

$$\mathbf{Pr}\left[B'=0\right]=\mathbf{Pr}\left[B'=1\right],$$

therefore B' is uniformly distributed over $\{0,1\}$ and, for all $b,b'\in\{0,1\}$,

$$\mathbf{Pr}\left[B=b\wedge B'=b'\right]=\frac{1}{4}.$$

We get the event $B=B'$ if and only if $b=b'$ and that happens in two out of four equally distributed cases, thus

$$\mathbf{Pr}\left[B=B'\right]=\frac{1}{2}.$$

□

Claim 1.4
Definition 1.1 ⇐ Definition 1.2.

Proof

This is easier to prove: assume that Π is a SKE scheme for which the Definition 1.1 does not hold; in particular, let $m_0, m_1 \in \mathcal{M}, c' \in \mathcal{C}$ be such that

$$\mathbf{Pr}\left[\mathrm{Enc}(K, m_0) = c'\right] > \mathbf{Pr}\left[\mathrm{Enc}(K, m_1) = c'\right].$$

Then, the adversary $\mathscr{A}$ can choose exactly m_0 and m_1 to challenge $\mathscr{C}$ and act as follows:

- if $\mathscr{C}$ outputs $C = c'$, $\mathscr{A}$ outputs $b' = 0$;
- if $\mathscr{C}$ outputs $C \neq c'$, $\mathscr{A}$ outputs $b' \leftarrow\$\{0,1\}$.

Now the probability of winning the game is

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1\right] = \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \wedge C = c'\right] + \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \wedge C \neq c'\right],$$

where

$$\begin{aligned}\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \wedge C \neq c'\right] &= \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \;\middle|\; C \neq c'\right]\mathbf{Pr}\left[C \neq c'\right] \\ &= \frac{1}{2}\mathbf{Pr}\left[C \neq c'\right]\end{aligned}$$

and

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \wedge C = c'\right] = \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \;\middle|\; C = c'\right]\mathbf{Pr}\left[C = c'\right].$$

Similarly,

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0\right] = \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \wedge C = c'\right] + \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \wedge C \neq c'\right],$$

where

$$\begin{aligned}\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \wedge C \neq c'\right] &= \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \;\middle|\; C \neq c'\right]\mathbf{Pr}\left[C \neq c'\right] \\ &= \frac{1}{2}\mathbf{Pr}\left[C \neq c'\right]\end{aligned}$$

and

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \wedge C = c'\right] = \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \;\middle|\; C = c'\right]\mathbf{Pr}\left[C = c'\right].$$

Now, if we compute the difference we obtain

$$\begin{aligned}&\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1\right] - \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0\right] \\ &\quad= \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \;\middle|\; C = c'\right]\mathbf{Pr}\left[C = c'\right] \\ &\qquad- \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \;\middle|\; C = c'\right]\mathbf{Pr}\left[C = c'\right] \\ &\quad= \left(\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1 \;\middle|\; C = c'\right] - \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0 \;\middle|\; C = c'\right]\right)\mathbf{Pr}\left[C = c'\right] \\ &\quad= \left(\mathbf{Pr}\left[\mathrm{Enc}(K, m_0) = c'\right] - \mathbf{Pr}\left[\mathrm{Enc}(K, m_1) = c'\right]\right)\mathbf{Pr}\left[C = c'\right] \\ &\quad> 0,\end{aligned}$$

therefore

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1\right] > \mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 0\right]$$

and, being the sum equal to 1,

$$\mathbf{Pr}\left[\textsc{Game}^{\mathrm{eav}}_{\Pi,\mathscr{A}} = 1\right] > \frac{1}{2}.$$

□

Applying both Claim 1.3 and Claim 1.4, we can conclude that Definition 1.1 and Definition 1.2 are equivalent.

2 Universal Hashing

Definition 2.1
A family $\mathcal{H} = \{h_s : X \to Y\}_{s \in S}$ of hash functions is called **t-wise independent** if for all sequences of distinct $x_1, \ldots, x_t \in X$ and for any sequence $y_1, \ldots, y_t \in Y$,

$$\mathbf{Pr}\left[h_s(x_1) = y_1 \wedge \ldots \wedge h_s(x_t) = y_t \mid s \leftarrow \$S\right] = \frac{1}{|Y|^t}.$$

Proposition 2.2
For any $t \geq 2$, if $\mathcal{H}$ is t-wise independent, then it is also $(t-1)$-wise independent.

Proof
Let $\mathcal{H}$ be t-wise independent and fix a sequence of distinct $x_1, \ldots, x_{t-1} \in X$ and a sequence $y_1, \ldots, y_{t-1} \in Y$. Fix $x \in X \setminus \{x_1, \ldots, x_{t-1}\}$ and write

$$\begin{aligned}
&\mathbf{Pr}\left[h_S(x_1) = y_1 \wedge \ldots \wedge h_S(x_{t-1}) = y_{t-1}\right] \\
&\quad = \sum_{y \in Y} \mathbf{Pr}\left[h_S(x_1) = y_1 \wedge \ldots \wedge h_S(x_{t-1}) = y_{t-1} \wedge h_S(x) = y\right] \\
&\quad = \sum_{y \in Y} \frac{1}{|Y|^t} \\
&\quad = \frac{|Y|}{|Y|^t} \\
&\quad = \frac{1}{|Y|^{t-1}}.
\end{aligned}$$

Then, the Proposition holds because x is arbitrary in $X \setminus \{x_1, \ldots, x_{t-1}\}$. □

Proposition 2.3
Let q be a prime and let $n \geq 2$ such that $q \geq n$. Then, the family

$$\mathcal{H}_n = \left\{h_s : \mathbb{Z}_q \to \mathbb{Z}_q : x \mapsto s_0 + s_1 x + \ldots + s_{n-1} x^{n-1}\right\}_{s=(s_0,\ldots,s_{n-1}) \in \mathbb{Z}_q^n}$$

is n-wise independent.

To prove this easily, we need a definition and a theorem.

Definition 2.4
A square **Vandermonde matrix** is a matrix in the form

$$V_n = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix}.$$

In other words, V_n is a square Vandermonde matrix if it is square and every row forms a geometric progression starting from 1.

Theorem 2.5
Let V_n be a square Vandermonde matrix. Using the same notation as in Definition 2.4, if $x_0, \ldots, x_{n-1}$ are all distinct, then V_n is invertible[1].

[1] In particular, the inverse form is known, so V_n^{-1} is efficiently computable.

Let's prove this for $n = 3$.

Proof ($n = 3$)
For $n = 3$, the form of V_3 is

$$\begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix}$$

and

$$\begin{aligned} \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} &= \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 \\ 0 & x_2 - x_0 & x_2^2 - x_0^2 \end{pmatrix} \\ &= (x_1 - x_0)(x_2 - x_0) \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 0 & 1 & x_1 + x_0 \\ 0 & 1 & x_2 + x_0 \end{pmatrix} \\ &= (x_1 - x_0)(x_2 - x_0) \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 0 & 1 & x_1 + x_0 \\ 0 & 0 & x_2 - x_1 \end{pmatrix} \\ &= (x_1 - x_0)(x_2 - x_0)(x_2 - x_1) \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 0 & 1 & x_1 + x_0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= (x_1 - x_0)(x_2 - x_0)(x_2 - x_1). \end{aligned}$$

Finally, if x_0, x_1, x_2 are all distinct, $\det(V_3) \neq 0$ and V_3 is invertible. □

Proof (Proposition 2.3)
Given $h_s \in \mathcal{H}_n, s = (s_0, \ldots, s_{n-1})$, we can easily evaluate h_s in $x_0, \ldots, x_{n-1} \in \mathbb{Z}_q$ simultaneously by constructing the matrix

$$V_n = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix}$$

and calculating the product

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} s_0 + s_1 x_0 + \ldots + s_{n-1} x_0^{n-1} \\ \vdots \\ s_0 + s_1 x_{n-1} + \ldots + s_{n-1} x_{n-1}^{n-1} \end{pmatrix} = \begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} s_0 \\ \vdots \\ s_{n-1} \end{pmatrix}$$

or, in short notation,

$$y = V_n \cdot s$$

If $x_0, \ldots, x_{n-1} \in \mathbb{Z}_q$ are all distinct, V_n is invertible by Theorem 2.5, thus we can write

$$s = V_n^{-1} y.$$

In particular, V_n represents a bijective map from $\mathbb{Z}_q^n$ to itself and we can rewrite the event

$$h_S(x_0) = y_0 \wedge \ldots \wedge h_S(x_{n-1}) = y_{n-1}$$

as

$$V_n S = y$$

or

$$S = V_n^{-1} y.$$

Finally, being S uniform over $\mathbb{Z}_q^n$ and being every y mapped into exactly one $z = V_n^{-1} y \in \mathbb{Z}_q^n$ and vice versa, we have that

$$\mathbf{Pr}\left[h_S(x_0) = y_0 \wedge \ldots \wedge h_S(x_{n-1}) = y_{n-1}\right] = \mathbf{Pr}\left[S = V_n^{-1} y\right] = \mathbf{Pr}\left[S = z\right] = \frac{1}{|\mathbb{Z}_q|^n}.$$

□

Corollary
Let $q \geq 3$ be a prime. Then, the family

$$\mathcal{H}_3 = \left\{h_s : \mathbb{Z}_q \to \mathbb{Z}_q : x \mapsto s_0 + s_1 x + s_2 x^2\right\}_{s=(s_0,s_1,s_2)\in\mathbb{Z}_q^3}$$

is 3-wise independent.

Proof
It is a particular case of Proposition 2.3 with $n = 3$. □

min-entropy.
Let $X \in \{0,1\}^n$ be a (k,n)-source, let $l = 128$ and $\varepsilon = 2^{-80}$. By the Leftover Hash Lemma, if the min-entropy of X is at least k,

$$H_\infty(X) \geq k \geq l + 2\log\left(\frac{1}{\varepsilon}\right) - 2 = 128 + 2 \cdot 80 - 2 = 286,$$

so X has a min-entropy of 286 bits and an entropy loss of $2\log(1/\varepsilon) - 2 = 158$ bits.

Suppose now that $k = 238$; the maximal amount of uniform randomness we can extract from X with statistical error $\varepsilon = 2^{-80}$ is

$$l \geq k - \left(2\log\left(\frac{1}{\varepsilon}\right) - 2\right) = 238 - 158 = 80$$

bits.
Missing: extension to 320 bits with computational assumptions.

3 One-Way Functions

Warning: I got 23/25, but I don't remember what was wrong here (I assume something imprecise or not well explained).
One-wayness of a PRG Let $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ be a PRG with one-bit stretch and prove that G is a OWF.

Use reduction to show that if G is not a OWF then it is not a PRG.
Suppose that a PPT adversary $\mathscr{A}^{\text{owf}}$ can invert G with non negligible probability (i.e., G is not a OWF). In particular, if the input to $\mathscr{A}^{\text{owf}}$ is $y = G(x)$, the output will be x' such that $G(x') = G(x)$ with non negligible probability and if $y \notin G\left(\{0,1\}^\lambda\right)$, the output will be a symbol $\square \notin \{0,1\}^\lambda$ with non negligible probability.
Now construct an adversary $\mathscr{A}^{\text{prg}}$ as follows: redirect every output from the challenger to $\mathscr{A}^{\text{owf}}$. If $\mathscr{A}^{\text{owf}}$ outputs some $x \in \{0,1\}^\lambda$, then $\mathscr{A}^{\text{prg}}$ outputs "PRG" to the challenger (i.e. he thinks that the number is generated from G); otherwise, $\mathscr{A}^{\text{prg}}$ outputs "RAND" to the

challenger (i.e. he thinks that the number is truly random).

If the z received from the challenger is an output from G, $\mathscr{A}^{\text{owf}}$ can invert G with non negligible probability finding some $x \in \{0,1\}^\lambda$, then $\mathscr{A}^{\text{prg}}$ can guess it to be an output from G with non negligible probability. Otherwise, z is truly random and can be inverted only about half of the times:

$$\mathbf{Pr}\left[z \in G\left(\{0,1\}^\lambda\right) \mid z \leftarrow\$\{0,1\}^{\lambda+1}\right] = \frac{\#G\left(\{0,1\}^\lambda\right)}{\#\{0,1\}^{\lambda+1}} = \frac{1}{2} + \nu(\lambda)$$

with $\nu \in \text{negl}(\lambda)$ (otherwise, $G(U_\lambda)$ could be distinguished from $U_{\lambda+1}$).
In conclusion, the output $z = G(x)$ can be guessed with probability near to $\frac{1}{p(\lambda)}$ and the output $z \leftarrow\$\{0,1\}^{\lambda+1}$ can be confused with $z = G(x)$ with probability near to $\frac{1}{2p(\lambda)}$ and, being $\frac{1}{p(\lambda)} - \frac{1}{2p(\lambda)}$ still not negligible, G would not be a PRG.

Therefore, an adversary $\mathscr{A}^{\text{owf}}$ cannot exist and G is also a OWF.

Non-secrecy of a OWF On the other hand, let $f : \{0,1\}^n \to \{0,1\}^n$ be a OWF: then the function

$$g : \{0,1\}^{n+\log(n)} \to \{0,1\}^{n+\log(n)+1} : (x, \langle j \rangle) \mapsto (f(x), j, x_j)$$

is also a OWF: given $g(x)$, if a PPT adversary $\mathscr{A}$ could invert g finding $(x', \langle j' \rangle)$ such that $g(x, \langle j \rangle) = g(x', \langle j' \rangle)$, $\mathscr{A}$ would have found in particular x' such that $f(x') = f(x)$, inverting the OWF f.

Now that we constructed the OWF g, for all $i \in \{1, \ldots, n + \log(n)\}$ we can construct an algorithm $\mathscr{A}_i$ as follows:

1. $\mathscr{A}_i$ gets the challenge to find $x' = x \,||\, \langle j \rangle$ given $g(x')$;
2. if $i > n$, $\mathscr{A}_i$ outputs $(g(x'))_i = j_{i-n} = x'_i$;
3. if $i \leq n$ but $i = j$, $\mathscr{A}_i$ outputs $(g(x'))_{n+\log(n)+1} = x_j = x_i$;
4. if $i \leq n$ and $i \neq j$, $\mathscr{A}_i$ outputs $b \leftarrow\$(\{0,1\})$.

Thus, for $i > n$, the probability to win for $\mathscr{A}_i$ is exactly 1; on the other hand, for $i \leq n$ we have

$$\begin{aligned}
&\mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle J \rangle)) = x'_i\right] \\
&\quad = \mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle J \rangle)) = x'_i \wedge i = J\right] + \mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle J \rangle)) = x'_i \wedge i \neq J\right] \\
&\quad = \mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle i \rangle)) = x'_i\right] \mathbf{Pr}\left[J = i\right] + \mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle J \rangle)) = x'_i \mid i \neq J\right] \mathbf{Pr}\left[J \neq i\right] \\
&\quad = 1 \cdot \frac{1}{2^{\log(n)}} + \frac{1}{2}\frac{2^{\log(n)} - 1}{2^{\log(n)}} \\
&\quad = \frac{1}{n} + \frac{1}{2} - \frac{1}{2}\frac{1}{n} \\
&\quad = \frac{1}{2} + \frac{1}{2n}.
\end{aligned}$$

Putting the two together, for all $i \in \{1, \ldots, n + \log(n)\}$ it is possible to construct a PPT algorithm $\mathscr{A}_i$ such that

$$\mathbf{Pr}\left[\mathscr{A}_i(g(X \,||\, \langle J \rangle)) = x'_i\right] \geq \frac{1}{2} + \frac{1}{2n},$$

thus breaking the randomicity of g (i.e. it is not possible to claim that every OWF hides at least one specific bit of the input).

4 Pseudorandom Generators

Let $G_1, G_2 : \{0,1\}^\lambda \to \{0,1\}^{\lambda+l}$ be two deterministic functions mapping λ bits into $\lambda + l$ bits ($l \geq \lambda + 1$) and let at least one of G_1, G_2 be a secure PRG. Then, we can construct a secure PRG $G^* : \{0,1\}^{2\lambda} \to \{0,1\}^{\lambda+l}$ by putting

$$G^*(x_1, x_2) = G_1(x_1) \oplus G_2(x_2).$$

In fact, if G_1 is a PRG, then

$$G^*(U_\lambda, U'_\lambda) = G_1(U_\lambda) \oplus G_2(U'_\lambda) \approx_c U_{\lambda+l} \oplus Y \approx_c U_{\lambda+l},$$

where $Y = G_2(U'_\lambda)$. Otherwise, G_2 must be a PRG and

$$G^*(U_\lambda, U'_\lambda) = G_1(U_\lambda) \oplus G_2(U'_\lambda) \approx_c X \oplus U_{\lambda+l} \approx_c U_{\lambda+l},$$

where $X = G_1(U'_\lambda)$.
We cannot improve this particular construction to get a λ bit seed PRG because we don't know anything about the relation between G_1 and G_2; in particular, if $G_1 = G_2$ (or, more in general, G_1 and G_2 have some bits in common), we get $G^*(x) = G_1(x) \oplus G_2(x) = 0$ (or, more in general, the bits that G_1 and G_2 have in common will be 0 after the XOR).

5 Pseudorandom Functions

This answer contains only the idea of the solution because I am late for the due.

Missing: first part of the exercise is poorly explained.
Let $\mathcal{F}$ be a PRF family and consider a computationally unbounded distinguisher $\mathscr{D}$. In particular, $\mathscr{D}$ "knows" (can precompute) $f_k(x)$ for every $x \in \{0,1\}^n$, for every$k \in \{0,1\}^\lambda$ and for every $f_k \in \mathcal{F}$.
The idea is that with a polynomial number of queries, $\mathscr{D}$ can restrict the domain (roughly halfing it) and try to guess k if $\mathscr{C}$ is playing the $\textsc{Real}_{\mathcal{F},\mathscr{D}}$ game, thus answering Real; otherwise (i.e. if $\mathscr{D}$ could not find a k), $\mathscr{D}$ answers Rand.

1. Let G be a λ to $\lambda+l$ PRG and let G' be the truncation of the output from G such that $G' : \{0,1\}^\lambda \to \{0,1\}^\lambda$. Then, we can ask to compute $F_k(0^\lambda) = G'(k) \oplus 0^\lambda = G'(k)$ and then, for every other message m, $F_k(m) = G'(k) \oplus m = F_k(0^\lambda) \oplus m$ and, knowing $F_k(0^\lambda)$, we can recover the message m and distinguish F_k from a true random function.

2. Let $F_s : \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a PRF for every $s \in \{0,1\}^\lambda$ and consider $F_x(k)$. The idea is to construct a particular PRF family properly indexed to show that $F_k(x)$ is secure but $F_x(k)$ is not, thus making not secure the construction.

3. Let $F'_k(x) = F_k(x \,||\, 0) \,||\, F_k(x \,||\, 1)$. Trying to break F'_k consists in trying to break both $F_k(\cdot \,||\, 0)$ and $F_k(\cdot \,||\, 1)$, thus if a distinguisher $\mathscr{D}'$ for F'_k exists, we can construct a distinguisher $\mathscr{D}$ for F_k as follows:

   (a) $\mathscr{D}'$ sends x to $\mathscr{D}$;
   (b) $\mathscr{D}$ sends $x \,||\, 0$, receiving z_0, and $x \,||\, 1$, receiving z_1, to $\mathscr{C}$;
   (c) $\mathscr{D}$ computes $z = z_0 \,||\, z_1$ and returns it to $\mathscr{D}'$;
   (d) this is repeated a polynomial number of time;
   (e) finally, $\mathscr{D}'$ responds either Real or Rand and $\mathscr{D}$ redirects the answer to $\mathscr{C}$.

   Now if $\mathscr{D}'$ can guess with non negligible probability between F'_k and random, $\mathscr{D}$ can guess with non negligible probability between F_k and random, thus breaking the PRF F_k. Therefore, this construction is secure.

6 Secret Key Encryption

Both CBC and OFB have the same problem: flipping one bit in one ciphertext block, say c_k, can compromise the plaintext block m_k and, eventually, m_{k+1}, but m_{k+2} and all the following blocks will remain intact.

Let $m = (m^0, m^1, m^2)$, let $k \leftarrow \$\{0,1\}^\lambda$ and let $h = 10^{n-1}$ (i.e. a block in which only the first bit is set to 1).

Compute CBC.
Generate a random Initialization Vector $c_0 \leftarrow \$\{0,1\}^n$ and compute

$$c_1 = P_k(c_0 \oplus m_1), \qquad c_2 = P_k(c_1 \oplus m_2), \qquad c_3 = P_k(c_2 \oplus m_3).$$

Suppose an attacker decides to alter the ciphertext as follows:

$$c'_0 = c_0, \qquad c'_1 = c_1 \oplus h, \qquad c'_2 = c_2, \qquad c'_3 = c_3.$$

Then, the decryption algorithm computes

$$m'_1 = P_k^{-1}(c'_1) \oplus c'_0, \qquad m'_2 = P_k^{-1}(c'_2) \oplus c'_1, \qquad m'_3 = P_k^{-1}(c'_3) \oplus c'_2,$$

that is

$$m'_1 = P_k^{-1}(c_1 \oplus h) \oplus c_0, \qquad m'_2 = P_k^{-1}(c_2) \oplus c_1 \oplus h, \qquad m'_3 = P_k^{-1}(c_3) \oplus c_2.$$

Now, m'_1 is completely destroyed since P_k is a PRP, but $m'_2 = m_2 \oplus h$ and $m'_3 = m_3$.

CCA to CBC.
It suffices to generate the two messages $m_0^* = (0^n, 0^n, 0^n)$ and $m_1^* = (1^n, 1^n, 1^n)$ to get the ciphertext c^* from the challenger; then, ask the challenger to decrypt $c' = c^* \oplus (h, 0^n, 0^n)$. Finally, choose b' corresponding to the message $m_{b'}^*$ that has the third block unchanged (i.e. choose $b' = 0$ if $c' = (\cdot, \cdot, 0^n)$ and $b' = 1$ otherwise).

Compute OFB.
Generate a random Initialization Vector $c_0 \leftarrow \$\{0,1\}^n$ and compute

$$c_1 = m_1 \oplus F_k(c_0), \qquad c_2 = m_2 \oplus F_k(F_k(c_0)).$$

Suppose an attacker decides to alter the ciphertext as follows:

$$c'_0 = c_0, \qquad c'_1 = c_1 \oplus h, \qquad c'_2 = c_2.$$

Then, the decryption algorithm computes

$$m'_1 = c'_1 \oplus F_k(c'_0), \qquad m'_2 = c'_2 \oplus F_k(F_k(c'_0)),$$

that is

$$m'_1 = c_1 \oplus h \oplus F_k(c'_0), \qquad m'_2 = c_2 \oplus F_k(F_k(c_0)).$$

Now we have $m'_1 = m_1 \oplus h$, that is m_1 with one bit flipped, and $m'_2 = m_2$.

CCA to OFB.
It suffices to generate the two messages $m_0^* = (0^n, 0^n)$ and $m_1^* = (1^n, 1^n)$ to get the ciphertext c^* from the challenger; then, ask the challenger to decrypt $c' = c^* \oplus (h, 0^n)$. Finally, choose b' corresponding to the message $m_{b'}^*$ that has the second block unchanged (i.e. choose $b' = 0$ if $c' = (\cdot, 0^n)$ and $b' = 1$ otherwise).

7 Message Authentication

Let $\Pi = (\text{Tag}, \text{Vrfy})$ be a MAC and consider the following game:
$\underline{\textsc{Game}^{\text{uf-cmva}}_{\Pi,\mathscr{A}}(1^\lambda) : \mathscr{A} \longrightarrow \mathscr{C}}$

1. $\mathscr{C}$ chooses a random key $k \leftarrow \$\mathcal{K}$.
2. $\mathscr{A}$ can query any (polynomial in λ) number of times $\mathscr{C}$ in the following way:
   - $\mathscr{A}$ can ask $\mathscr{C}$ to tag a particular message m (i.e. can ask $\text{Tag}_k(m)$);
   - $\mathscr{A}$ can ask $\mathscr{C}$ to verify a particular couple (m, ϕ) (i.e. can ask $\text{Vrfy}_k(m, \phi)$).
3. $\mathscr{A}$ then decides to send (forge) a *fresh* couple (m^*, ϕ^*) to $\mathscr{C}$.
4. The output is 1 if and only if $\text{Vrfy}_k(m^*, \phi^*) = 1$.

Definition 7.1
A MAC Π has **unforgeability under chosen message and verification attacks** if

$$\mathbf{Pr}\left[\textsc{Game}^{\text{uf-cmva}}_{\Pi,\mathscr{A}}(1^\lambda)\right] \leq \nu(\lambda),$$

where $\mathscr{A}$ is a PPT adversary and ν is a negligible function in λ.

Theorem 7.2
If Π has unique tags (i.e. for every key k there is only one valid tag ϕ for each message m), then UF-CMA and UF-CMVA are equivalent.

Proof
The implication UF-CMVA $\Rightarrow$ UF-CMA is trivial; let's prove that UF-CMA $\Rightarrow$ UF-CMVA under unique tag assumption.
Suppose Π to be UF-CMA but not UF-CMVA and suppose an adversary $\mathscr{A}$ trying to break Π playing $\textsc{Game}^{\text{uf-cma}}_{\Pi,\mathscr{A}}$. Use reduction: consider $\mathscr{A}^v$ able to break Π playing $\textsc{Game}^{\text{uf-cmva}}_{\Pi,\mathscr{A}^v}$ and interfacing with $\mathscr{A}$:

$$\mathscr{A}^v \longleftrightarrow \mathscr{A} \longleftrightarrow \mathscr{C}^{\text{uf-cma}}.$$

Let $\mathscr{A}^v$ make all the queries:

- if $\mathscr{A}^v$ asks for a Tag query, m is sent from $\mathscr{A}^v$ to $\mathscr{A}$, which redirects m to $\mathscr{C}$ obtaining $\phi = \text{Tag}_k(m)$ and sending it back to $\mathscr{A}^v$;
- if $\mathscr{A}^v$ asks for a Vrfy query, (m, ϕ) is sent from $\mathscr{A}^v$ to $\mathscr{A}$, which sends m to $\mathscr{C}$ obtaining ϕ'; then, $\mathscr{A}$ compares ϕ and ϕ' and responds 1 to $\mathscr{A}^v$ if and only if $\phi = \phi'$. Note that this is the only operation to do to verify a tag because of tag uniqueness.

Now, $\mathscr{A}^v$ should be able to forge a tag ϕ^* for a message m^* such that (m^*, ϕ^*) is fresh and $\text{Vrfy}_k(m^*, \phi^*) = 1$ with non negligible probability; sending the pair (m^*, ϕ^*) to $\mathscr{A}$, the situation is that $\mathscr{A}$ forged a message/tag pair using $\mathscr{A}^v$ that is valid with non negligible probability, therefore Π cannot be UF-CMA.

Things are different if Π has not unique tags, i.e. if exists some message m such that ϕ_1, ϕ_2 are both valid tags but $\phi_1 \neq \phi_2$.

Example 7.3

Let $\Pi = (\text{Tag}, \text{Vrfy})$ be a MAC satisfying UF-CMA and construct a new MAC Π' as follows.
Let $\text{Tag}'_k(m) = \text{Tag}_k(m) \,||\, 0^{\log(\lambda)}$ and define Vrfy'_k:

- break $\phi' = \phi \,||\, \langle i \rangle$, where $\langle i \rangle$ is the binary representation of i;
- compute $d_0 = \text{Vrfy}_k(\phi)$;
- if $d_0 = 0$ or $i = 0$, put $d = d_0$;
- if $d_0 \neq 0$ and $i \neq 0$, put $d = k_j$, where k_j is the jth bit of k;
- let $\text{Vrfy}'_k(m, \phi')$ output d.

Now try to break $\Pi' = (\text{Tag}', \text{Vrfy}')$.
As usual, show that Π' is UF-CMA by reduction: let $\mathscr{A}$ be a UF-CMA adversary against Π and let $\mathscr{A}'$ be a UF-CMA adversary against Π'. The situation is the following:

$$\mathscr{A}' \longleftrightarrow \mathscr{A} \longleftrightarrow \mathscr{C}.$$

For every message m queried by $\mathscr{A}'$, $\mathscr{A}$ asks $\mathscr{C}$ for a tag, appends $0^{\log(\lambda)}$ to the result and returns $\phi' = \phi \,||\, \langle 0 \rangle$ to $\mathscr{A}'$. Since these are the only queries available for $\mathscr{A}'$, the only other thing it can do is trying to forge a pair (m^*, ϕ^*) and sending it to $\mathscr{A}$. Note that to win the game, $\mathscr{A}'$ must forge a tag such that $d_0 = 1$; otherwise, the output d is 0 and the game is lost. The only possibility for d_0 to be 1 is that, given $\phi' = \phi \,||\, \langle 0 \rangle$, $\text{Vrfy}_k(m, \phi) = 1$, thus, if $\mathscr{A}'$ wins the game with non negligible probability, the same happens for $\mathscr{A}$ and the initial MAC Π is not UF-CMA.

To break Π' in a UF-CMVA game, it suffices for the adversary to query a message m obtaining a tag $\phi' = \phi \,||\, \langle 0 \rangle$ and then asking the challenger to verify the tags $(m, \phi \,||\, \langle i \rangle)$ for each $i \in \{1, \ldots, n-1\}$, obtaining d_i. Since the verification queries are only $n-1 \in \text{poly}(\lambda)$, the adversary can effectively do all these queries. In particular, the adversary obtains the bits $k_1, \ldots, k_{\lambda-1}$. Finally, he can guess the first bit k_0 trying to compute the tag for a fresh message m_0 and asking to verify $\phi'_0 = \text{Tag}_{k_0 || \ldots || k_{\lambda-1}}$.

Now the adversary knows the key and can forge all the message he wants with probability of success 1.

CBC-MAC.

Assume having a fixed length CBC-MAC that computes Tag using a PRF F_k as follows: given a message $m = (m_1, \ldots, m_t)$, compute recursively

$$\phi_i = F_k(m_i \oplus \phi_{i-1}), \qquad \text{where} \qquad \phi_0 = 0^n \qquad \text{and} \qquad i \in \{1, \ldots, t\}.$$

- This MAC can't be used to authenticate variable-length messages: given two messages (m_1) and (m_1, m_2) and their relative tags $\phi_1 = F_k(m_1)$ and $\phi_2 = F_k(m_2 \oplus F_k(m_1))$, it is possible to forge a tag $\phi^* = \phi_2$ for the message $m^* = m_2 \oplus \phi_1 = m_2 \oplus F_k(m_1)$.
- A variant in which ϕ_0 is randomized and output with the tag is not secure even for fixed length messages: one can ask a tag for the message $(m_1, \ldots, m_t)$, obtaining $(\phi_0, \ldots)$, and forge the tag $(\phi_0 \oplus m_1, \ldots)$ for the (fresh) message $(0^n, m_2, \ldots, m_t)$. Note that this security issue holds for both outputs (ϕ_0, ϕ_t) and $(\phi_0, \phi_1, \ldots, \phi_t)$.

Missing: I interpreted this wrong: in the third variant, ϕ_0 is not random but $\phi_0 = 0^n$. Anyway, this case is insecure too.

Cryptography—Homework 1*

Sapienza University of Rome
Master's Degree in Computer Science
Master's Degree in Cybersecurity
Master's Degree in Mathematics

Daniele Venturi

Due Date: November 13, 2019

1 Perfect Secrecy — 20 Points

(a) Prove or refute: An encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ with key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$ is perfectly secret if and only if the following holds: For every probability distribution M over $\mathcal{M}$, and every $c_0, c_1 \in \mathcal{C}$, we have $\Pr[C = c_0] = \Pr[C = c_1]$, where $C := \mathsf{Enc}(K, M)$ with K uniform over $\mathcal{K}$.

(b) Let $(\mathsf{Enc}, \mathsf{Dec})$ be a perfectly secret encryption scheme over message space $\mathcal{M}$ and key space $\mathcal{K}$, satisfying the following relaxed correctness requirement: There exists $t \in \mathbb{N}$ such that, for all $m \in \mathcal{M}$, it holds that $\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] \geq 2^{-t}$ (where the probability is over the choice of $k \leftarrow\!\!\$\, \mathcal{K}$). Prove that $|\mathcal{K}| \geq |\mathcal{M}| \cdot 2^{-t}$.

2 Universal Hashing — 20 Points

(a) A family $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \in \mathcal{S}}$ of hash functions is called t-wise independent if for all sequences of *distinct* inputs $x_1, \ldots, x_t \in \mathcal{X}$, and for any output sequence $y_1, \ldots, y_t \in \mathcal{Y}$ (not necessarily distinct), we have that:

$$\Pr[h_s(x_1) = y_1 \wedge \cdots \wedge h_s(x_t) = y_t : \ s \leftarrow\!\!\$\, \mathcal{S}] = \frac{1}{|\mathcal{Y}|^t}.$$

(i) For any $t \geq 2$, show that if $\mathcal{H}$ is t-wise independent, then it is also $(t-1)$-wise independent.

*Some of the exercises are taken from the book *"Introduction to Modern Cryptography"* (second edition), by Jonathan Katz and Yehuda Lindell.

(ii) Let q be a prime. Show that the family $\mathcal{H} = \{h_s : \mathbb{Z}_q \to \mathbb{Z}_q\}_{s \in \mathbb{Z}_q^3}$, defined by

$$h_s(x) := h_{s_0,s_1,s_2}(x) := s_0 + s_1 \cdot x + s_2 \cdot x^2 \bmod q$$

is 3-wise independent.

(b) Say that X is a (k, n)-source if $X \in \{0,1\}^n$, and the min-entropy of X is at least k. Answer the following questions:

(i) Suppose that $\ell = 128$; what is the minimal amount of min-entropy needed in order to obtain statistical error $\varepsilon = 2^{-80}$ when applying the leftover hash lemma? What is the entropy loss?

(ii) Suppose that $k = 238$; what is the maximal amount of uniform randomness that you can obtain with statistical error $\varepsilon = 2^{-80}$ when applying the leftover hash lemma? Explain how to obtain $\ell = 320$ using computational assumptions.

3 One-Way Functions 20 Points

(a) Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a PRG with λ-bit stretch. Prove that G is by itself a one-way function.

(b) Let $f : \{0,1\}^n \to \{0,1\}^n$ be a OWF. Consider the function $g : \{0,1\}^{n+\log n} \to \{0,1\}^{n+\log n+1}$ defined by $g(x||j) := (f(x), j, x_j)$, where $x := (x_1, \ldots, x_n)$, and j is interpreted as an integer in $[n]$ (i.e., $|j| = \log n$).

(i) Show that g is a OWF if f is.

(ii) Show that for every $i \in [n']$ there is a PPT algorithm $\mathcal{A}_i$ for which

$$\Pr\left[\mathcal{A}_i(g(x')) = x'_i : \ x' \leftarrow\!\!\$\ \{0,1\}^{n+\log n}\right] \geq \frac{1}{2} + \frac{1}{2n},$$

where $x' = (x_1, \ldots, x_{n'})$.

4 Pseudorandom Generators 20 Points

(a) Let $G_1, G_2 : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ be two deterministic functions mapping λ bits into $\lambda + \ell$ bits (for $\ell \geq 1$). You know that at least one of G_1, G_2 is a secure PRG, but you don't know which one. Show how to design a secure PRG $G^* : \{0,1\}^{2\lambda} \to \{0,1\}^{\lambda+\ell}$ by combining G_1 and G_2.

(b) Can you prove that your construction works when using the same seed $s^* \in \{0,1\}^\lambda$ for both G_1 and G_2? Motivate your answer.

5 Pseudorandom Functions — 25 Points

(a) Show that no PRF family can be secure against computationally unbounded distinguishers.

(b) Analyze the following candidate PRFs. For each of them, specify whether you think the derived construction is secure or not; in the first case prove your answer, in the second case exhibit a concrete counterexample.

(i) $F_k(x) = G'(k) \oplus x$, where $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ is a PRG, and G' denotes the output of G truncated to λ bits.

(ii) $F_k(x) := F_x(k)$, where $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\ell$ is a PRF.

(iii) $F'_k(x) = F_k(x||0)||F_k(x||1)$, where $x \in \{0,1\}^{n-1}$.

6 Secret-Key Encryption — 20 Points

(a) Prove that no secret-key encryption scheme $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ can achieve chosen-plaintext attack security in the presence of a computationally unbounded adversary (which thus can make an exponential number of encryption queries before/after being given the challenge ciphertext).

(b) Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a family of pseudorandom permutations, and define a fixed-length encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ as follows: Upon input message $m \in \{0,1\}^{n/2}$ and key $k \in \{0,1\}^\lambda$, algorithm Enc chooses a random string $r \leftarrow\!\!\$\, \{0,1\}^{n/2}$ and computes $c := F_k(r||m)$. Show how to decrypt, and prove that this scheme is CPA-secure for messages of length $n/2$.

7 Message Authentication — 25 Points

(a) Assume UF-CMA MACs exist. Prove that there exists a MAC that is UF-CMA but is not strongly UF-CMA. (Recall that strong unforgeability allows the attacker to produce a forgery (m^*, τ^*) such that $(m^*, \tau^*) \neq (m, \tau)$ for all messages m que)

(b) Assume a generalization of MACs where a MAC Π consists of a pair of algorithms $(\mathsf{Tag}, \mathsf{Vrfy})$, such that Tag is as defined in class (except that it could be randomized), whereas Vrfy is a deterministic algorithm that takes as input a candidate pair (m, τ) and returns a decision bit $d \in \{0,1\}$ (indicating whether τ is a valid tag of m). Consider a variant of the game defining UF-CMA security of a MAC $\Pi = (\mathsf{Tag}, \mathsf{Vrfy})$, with key space $\mathcal{K} = \{0,1\}^\lambda$, where the adversary is additionally granted access to a verification oracle $\mathsf{Vrfy}(k, \cdot, \cdot)$.

(i) Make the above definition precise, using the formalism we used in class. Call the new notion "unforgeability under chosen-message and verification attacks" (UF-CMVA).

(ii) Show that whenever a MAC has unique tags (i.e., for every key k there is only one valid tag τ for each message m) then UF-CMA implies UF-CMVA.

(iii) Show that if tags are not unique there exists a MAC that satisfies UF-CMA but not UF-CMVA.

5 Actively Secure ID Schemes — 30 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{P}, \mathsf{V})$ be an ID scheme. Informally, an ID scheme is actively secure if no efficient adversary A (given just the public key pk) can make V accept, even after A participates maliciously in poly-many interactions with P (where the prover is given both the public key pk and the secret key sk). More formally, we say that Π satisfies active security if for all PPT adversaries A there is a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that

$$\Pr\left[\mathbf{Game}^{\mathsf{mal\text{-}id}}_{\Pi,\mathsf{A}}(\lambda) = 1\right] \leq \nu(\lambda),$$

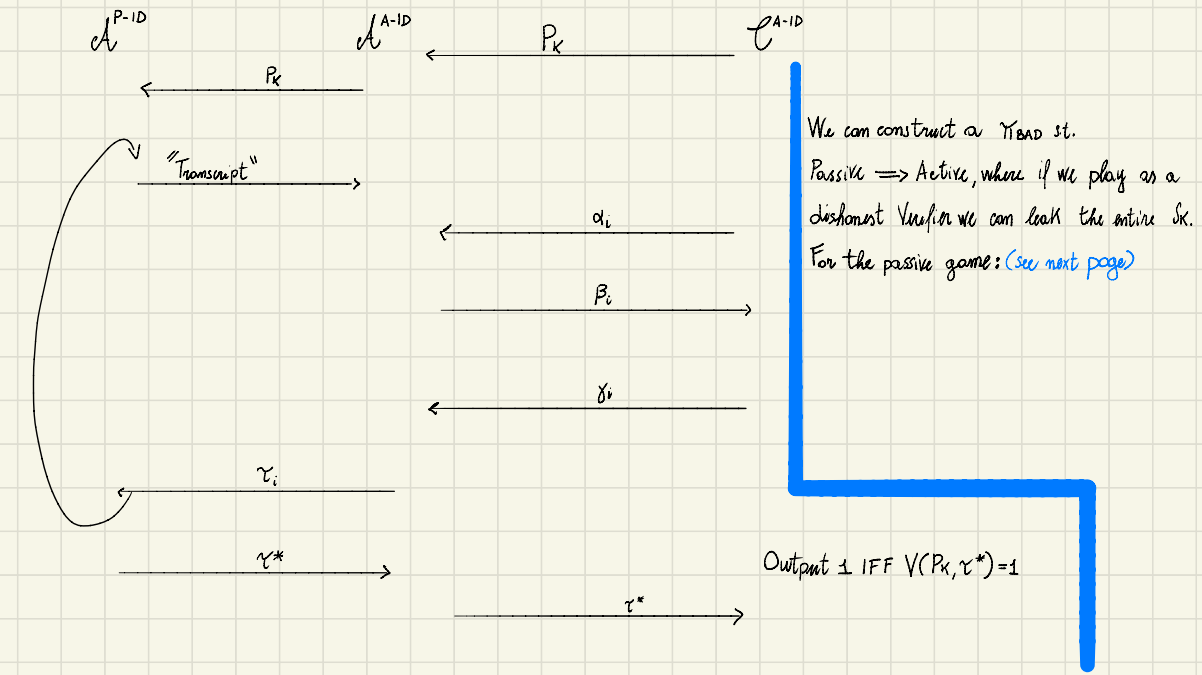
where the game $\mathbf{Game}^{\mathsf{mal\text{-}id}}_{\Pi,\mathsf{A}}(\lambda)$ is defined as follows:

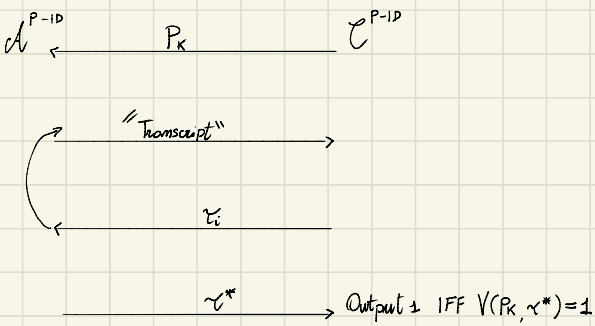
- The challenger runs $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$, and returns pk to A.
- Let $q(\lambda) \in \mathrm{poly}(\lambda)$ be a polynomial. For each $i \in [q]$, the adversary can run the protocol Π with the challenger (where the challenger plays the prover and the adversary plays the malicious verifier), obtaining transcripts $\tau_i \leftarrow\!\!\$\ (\mathsf{P}(pk, sk) \rightleftarrows \mathsf{A}(pk))$.
- Finally, the adversary tries to impersonate the prover in an execution of the protocol with the challenger (where now the challenger plays the honest verifier), yielding a transcript $\tau^* \leftarrow\!\!\$\ (\mathsf{A}(pk) \rightleftarrows \mathsf{V}(pk))$.
- The game outputs 1 if and only if the transcript τ^* is accepting, i.e. $\mathsf{V}(pk, \tau^*) = 1$.

Answer the following questions.

(a) Prove that passive security is strictly weaker than active security. Namely, show that every ID scheme Π that is actively secure is also passively secure, whereas there exists a (possibly contrived) ID scheme Π_{bad} that is passively secure but not actively secure.

Active $\Rightarrow$ Passive

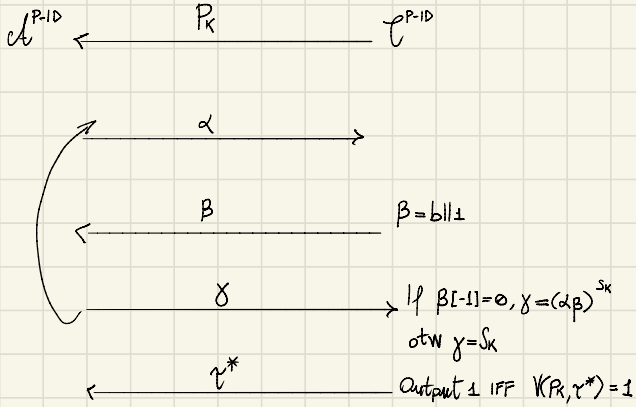




Now r_i will be as follows, from the point of view of $\mathcal{A}$:

- $\alpha \leftarrow U$
- $\beta = b||0$ where $b \leftarrow\$ U$
- $\gamma = B[-1](S_K)+(1+\beta[-1])(\alpha\beta)^{S_K}$

So the above 1D-scheme has still Passive Security. Instead for the Active Security we can play the following interaction:



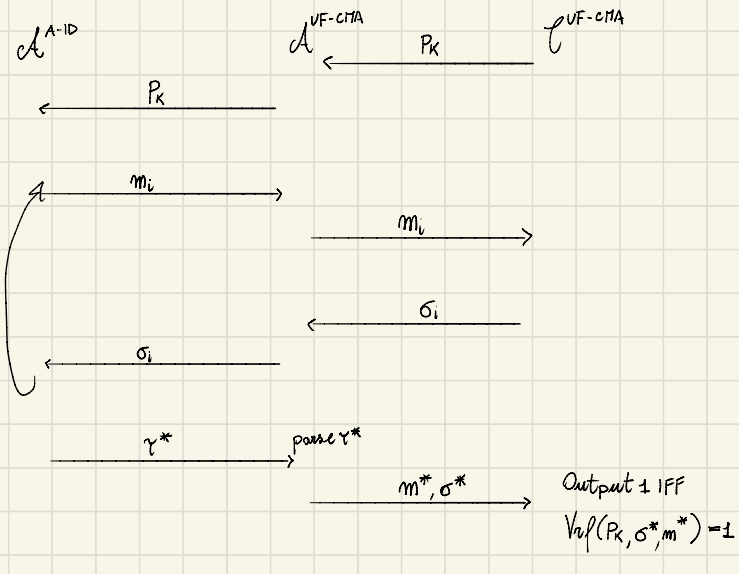
Now that we have S_K we can always create a valid r^*.

5 Actively Secure ID Schemes

(b) Let $\Pi' = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, with message space $\mathcal{M}$. Prove that if Π' is UF-CMA, the following ID scheme $\Pi = (\mathsf{KGen}, \mathsf{P}, \mathsf{V})$ (based on Π') achieves active security:

$\mathsf{P}(pk, sk) \rightleftarrows \mathsf{V}(pk)$: The verifier picks random $m \leftarrow\!\!\$\ \mathcal{M}$, and forwards m to the prover. The prover replies with $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk, m)$, and finally the verifier accepts if and only if $\mathsf{Vrfy}(pk, m, \sigma) = 1$.

Suppose $\exists A^{A\text{-}ID}$ which is able to break the active security of Π



(c) Is the above protocol honest-verifier zero-knowledge? Prove your answer.

The HVZK property comes from the fact that the signature scheme used in the following way:

- The verifier sends the message
- The Prover signs the message

doesn't leak any information about the secret used for signing the message (assuming V always as honest verifier). For the property of the signature it will never reveal anything about the secret used to generate σ.

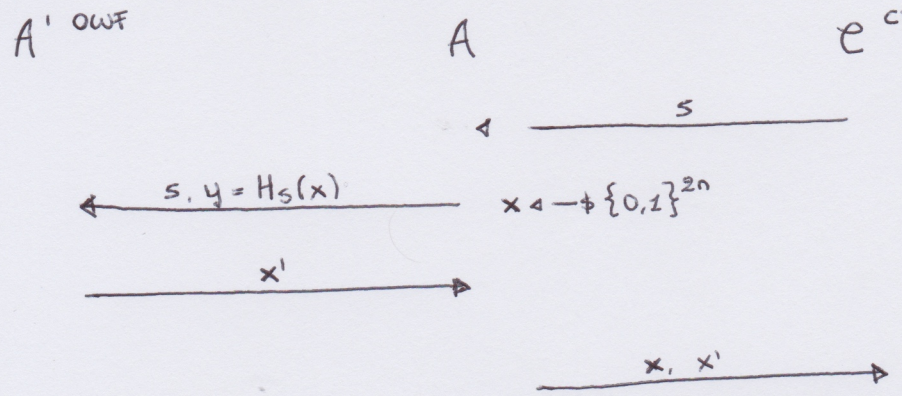
HOMEWORK 2 CRYPTOGRAPHY ALESSIA MARROCCO 1774884

EXERCISE 1

$\mathcal{H} = \{ H_s : \{0,1\}^{2n} \to \{0,1\}^n \}_{s \in \{0,1\}^2}$

a) i. $\mathcal{H}$ is CRHF $\Rightarrow$ $\mathcal{H}$ is OWF

To show this property, let's make a reduction



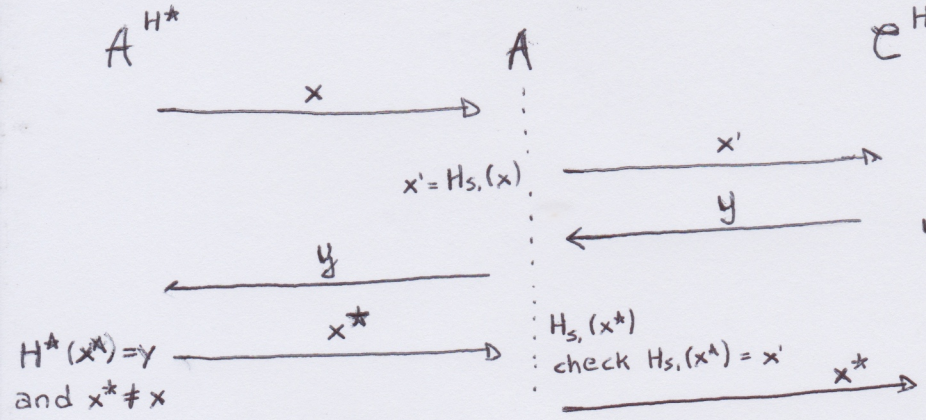
When does not A win?

Since CRHF game wants the final couple (x,x') with $x \neq x'$, if A'^{OWF} returns $x' = x$ the CRHF game doesn't work. This BAD events happens with

$$P[x = x'] = Col(X,X') = \sum_x P[X = x \wedge X' = x] = \sum_x P[X = x]\, P[X' = x] = \frac{1}{2^{2n}}$$

a) ii. Intuitively if a function is not compressing ($h: n \to n$) and is collision resistant, we can assume it will coincident to the best hash function possible (a bijective one). In this case the function will obviously be CRHF since it is impossible to find $x' \neq x$ such that $h(x') = h(x)$, however this will not be one way since there will be a unique correspondence between an element in the domain and an element in the codomain.

b) Given $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ with $H^*: 4n \to n$. Suppose $\exists A^{H^*}$ which is able to find a collission in H^*. Consider the following two games:



There is a BAD event in which A^{H^*} outputs a collision for H_{s_2}, meaning that $H_{s_1}(x) = H_{s_1}(x')$ in this case the second part of the reduction doesn't work. But $Pr[BAD]$ is negligible since H_{s_1} is collision resistant by definition. But now $H^*(x') = H^*(x)$ since x' was a collision for H^* but this must be a collision for H'_{s_2} which was a CRHF for hyphotesis.

EXERCISE 2

a) If we can compute g^{ab} given (g, g^a, g^b), then to compute g^{a^2} it's sufficies to have (g, g^a, g^a). On the other hand if we can compute g^{a^2}, then we can do the following:

$$g^{(a+b)^2} = g^{a^2+b^2+2ab} = g^{a^2}\, g^{b^2}\, g^{2ab}$$

then dividing by $g^{a^2}\, g^{b^2}$, we get g^{2ab} so $g^{ab} = \sqrt{g^{2ab}}$

b) Under the Discrete Log Assumption f is one way. In fact We Know that $f_{g,p}(x) = g^x \bmod p$, so it can be evaluated in poly time so:

f is ONE way $\iff \mathbb{P}[A(1^l, y) = x' ; x \leftarrow \{0,1\}^l, y = f_{g,p}(x), y = f_{g,p}(x')]$

$\iff \mathbb{P}[A(1^l, y) = x ; x \leftarrow \{0,1\}^l, y = g^x \bmod p]$

$\uparrow$

$f_{g,p}$ is a PERMUTATION, $f_{g,p}(x) = f_{g,p}(x') \Rightarrow x = x'$

$\iff$ the discrete log assumption hold.

CLAIM: x_b is the least significant bit of x

$\iff g' = \dfrac{g^x}{g^{x_b}} \in QR_p$ indeed $x = x_t 2^t + \cdots + x_1 \cdot 2 + x_b$

$\Rightarrow g' = g^{(x - x_b)} = g^{2(x_t 2^t + \cdots + x \cdot 2 + x_b - x_b)} = g^{2(x_t \cdot 2^{t-1} + \cdots + x)}$

$\Rightarrow g'$ is an even power of g' so $g' \in QR_p$

then I can build A against h in this way:

- if $y^{\frac{p-1}{2}} \equiv 1 \bmod p \Rightarrow$ A outputs 0
- if $y^{\frac{p-1}{2}} \not\equiv 1 \bmod p \Rightarrow$ A outputs 1

in fact:

- if $y^{\frac{p-1}{2}} \equiv 1$ then $y = g^x \in QR_p \Rightarrow y = \dfrac{g^x}{g^0} \in QR_p \Rightarrow x_b = 0$
- if $y^{\frac{p-1}{2}} \not\equiv 1$ then $y = g^x \notin QR_p \Rightarrow y = \dfrac{g^x}{g^0} \in QR_p \Rightarrow x_b = 1$

So $\mathbb{P}[A(f_{g,p}(x)) = h(x)] = 1 > \frac{1}{2} + \frac{1}{p(l)}$

c) Claim: let $x_i = x \bmod p_i$ $y_i = y \bmod p_i$, x solves $x^2 = y$ IFF

$$\begin{cases} x_1^2 \equiv y_1 \bmod p_1 \\ \vdots \\ x_5^2 = y_5 \bmod p_5 \end{cases} \quad (1)$$

Proof: $x^2 = y \bmod N \Rightarrow \exists K \in \mathbb{Z}$ such that $y = KN + x^2 = K p_1 \cdots p_5 + x^2$

So $\begin{cases} y_1 \equiv x_1^2 \bmod p_1 \\ y_5 \equiv x_5^2 \bmod p_5 \end{cases}$ We know by CRT that if $x_1, \dots, x_5$ are solutions of (1) then $\exists x$ s.t $x \equiv x_i \bmod p_i$, $i = 1, \dots, 5$

Then:

$$(1) \iff \begin{cases} y_1 - x_1^2 \equiv 0 \bmod p_1 \\ \vdots \\ y_5 - x_5^2 \equiv 0 \bmod p_5 \end{cases} \iff y \equiv x^2 \bmod N \text{ where } y \text{ is the only solution of } \begin{cases} y_1 \equiv y_1 \bmod p_1 \\ \vdots \\ y_5 \equiv y_5 \bmod p_5 \end{cases}$$

We know that the only solutions of $y_i \equiv x_i^2 \bmod p_i$ are $\pm x_i$, with $+x_i \neq -x_i$, since $p \neq 2$. So there are 2^5, in fact each solutions solves

$$\begin{cases} x = \pm x_1 \bmod p_1 \\ x = \pm x_5 \bmod p_5 \end{cases}$$

EXERCISE 3

a) Given $(P_K, S_K) \leftarrow \mathcal{G}en(1^2)$, $|Enc(p_K, b)| = O(\log m)$ $\forall b \in \{0,1\}$

in the considered case the length of the cyphertext is bounded logarithmically in the security parameter so the size of all cyphertexts is bounded by a polinomial.

So it holds that given $x \in \{0,1\}$, $|x| \leq \log m = \sum_{i=0}^{\log m} 2^i = 2^{\log m + 1} - 1 = 2m - 1 = p(n)$

Now consider the following adversary A in the CPA-game:

- given p_K, A computes $c_0 = Enc(p_K, 0)$, $c_1 = Enc(p_K, 1)$
- then A outputs $m_0 = 0$, $m_1 = 1$
- A checks c, when it comes from the challenger, and A outputs 0 or 1 if c is c_0 or c_1, otherwise A outputs a random bit.

A wins with non-negligible probability.

Proof. Let $C(b)$ denote the set of all possible cyphertexts of the message b. Whereas the cyphertext size is bounded log, there exists a poly p s.t $|C(b)| \leq p(n)$

So, there exists $c_b^* \in C(b)$ s.t.

$$\mathbb{P}[A \text{ outputs } 0 \mid c \leftarrow Enc(p_K, 0)] = \mathbb{P}[A \text{ outputs } 1 \mid c \leftarrow Enc(p_K, 1)] \geq$$

$$1 - \mathbb{P}[c = c_0] + \frac{1}{2}\mathbb{P}[c \neq c_0] = \frac{1}{2}\mathbb{P}[c = c_0] + \frac{1}{2}\mathbb{P}[c = c_0] + \frac{1}{2}\mathbb{P}[c \neq c_0] =$$

$$\frac{1}{2}\mathbb{P}[c = c_0] + \frac{1}{2} \geq \frac{1}{2}\mathbb{P}[c = c_0 = c_0^*] + \frac{1}{2} \geq \frac{1}{2} \cdot \frac{1}{p(n)^2} - \frac{1}{2}$$

Concluding

$$\mathbb{P}[GAME_{A,\Pi}^{CPA}(n) = 1] = \frac{1}{2} \cdot \mathbb{P}[A \text{ outputs } 1 \mid c \leftarrow Enc(p_K, 1)] + \frac{1}{2}\mathbb{P}[A \text{ outputs } 0 \mid c \leftarrow Enc(p_K, 0)] \geq$$

$$\frac{1}{2} \cdot 2 \cdot \left(\frac{1}{2} \cdot \frac{1}{p(n)^2} + \frac{1}{2}\right) = \frac{1}{2}\frac{1}{p(n)^2} + \frac{1}{2} \quad \text{that is not-negligible}$$

So, to achieve CPA-security we need the "superlogarithm" length of the cyphertext.

For example given $(p_K, s_K) \leftarrow \mathcal{G}en(1^2)$, $Enc(p_K, b) = \omega \log n$.

In this case we have $2^{c \log m}$ n^c possible cyphertexts with n^c possibilities ~~xxxxxxx~~
A wins with negl prob infact.

$$\mathbb{P}[GAME^{CPA}_{\pi,A}(n) = 1] = \frac{1}{2}\mathbb{P}[A \text{ outputs } 1 \mid c \leftarrow Enc(pk,1)] + \frac{1}{2}\mathbb{P}[A \text{ outputs } 0 \mid c \leftarrow Enc(pk,0)] \geq$$

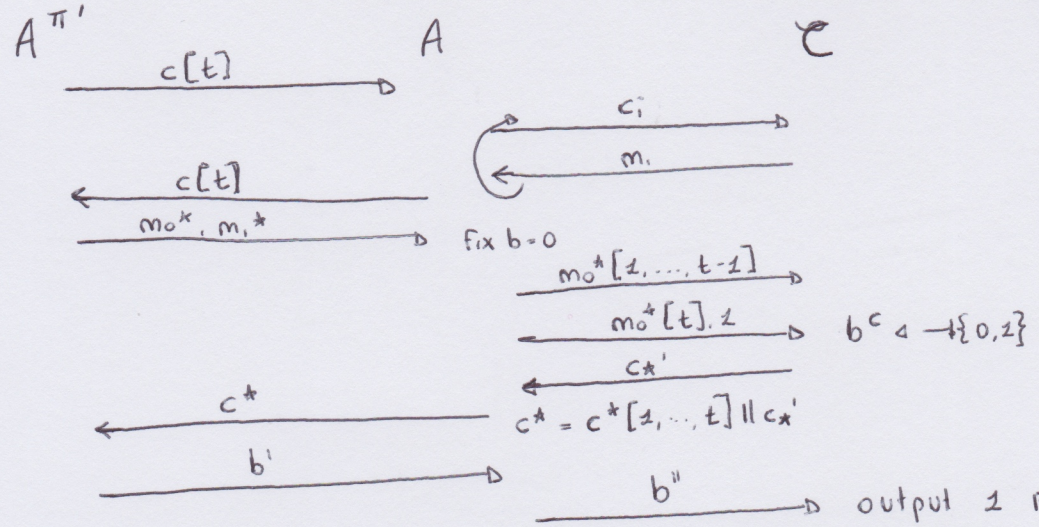
$$\frac{1}{2} \cdot 2\left(\frac{1}{2} \cdot \frac{1}{(n^c)^2} + \frac{1}{2}\right) = \frac{1}{2} + \frac{1}{n(k)^2} \Rightarrow \text{negligible}$$

b) i. We have to demonstrate if Π is CCA 1 $\Rightarrow \Pi'$ is also CCA 1, in order to do this observe the following reduction scheme:

Suppose $\exists A^{\Pi'}$ which is able to break the CCA 1 security of Π' _ $A^{\Pi'}$ sends ciphertext composed by t elements _ A takes every single element and sends to C to get the plaintext of each one _ Then recombines the plain text and sends back the single plaintext to $A^{\Pi'}$ _

At the start of the challenge $A^{\Pi'}$ sends m_0, m_1 of t bits to A _ A sends to C the first t-1 bytes of m_0 and receives the corresponding t-1 cyphertexts then A sends to C the challenge as: $m_0[t]$ and 1, recewing the cyphertext c_*' of one of the two _ At this point A recombines all of t-1 cyphertexts + the last one c_*' and sends back to A^{Π} _ Now A will just forward the response _

The probability will be $\mathbb{P}[A^{\Pi} = 0 \mid b^c = 0] - \mathbb{P}[A^{\Pi} = 1 \mid b^c = 1] = \frac{1}{2} + negl(\lambda) - \frac{1}{2} \geq negl(\lambda)$



ii. Π CCA 2 $\Rightarrow \Pi' \neg$ CCA2

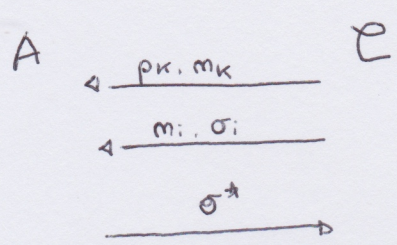
Consider the following PKE - scheme:

- $Enc(P_K, m[t]) = Enc(pk, m_1) \| \cdots \| Enc(pk, m_t)$
- $Dec(S_K, c[t]) = Dec(sk, c_1) \| \cdots \| Dec(sk, c_t)$

Since in CCA2 I can make decryption queries after the challenge, I can create a $c' \neq c^*$ just by inverting the first two bits of c^* ($c^* = c_1^* \| c_2^* \| \cdots \| c_t^*$) now $c' = c_2^* \| c_1^* \| \cdots \| c_t^*$ _ Now when I received the decrypted message I can simply switch the first two bits again and discover which of the two challenge messages was encrypted _

EXERCISE 4

a) $GAME^{RUF-RMA}_{\Pi,A}(\lambda)$



$(pk, sk) \leftarrow\!\$\ KGen(1^\lambda)$

$m^* \leftarrow\!\$\ \mathbb{Z}_m^*$

$m_i \leftarrow\!\$\ \mathbb{Z}_m^*$ s.t $m^* \notin \{m_i\}$

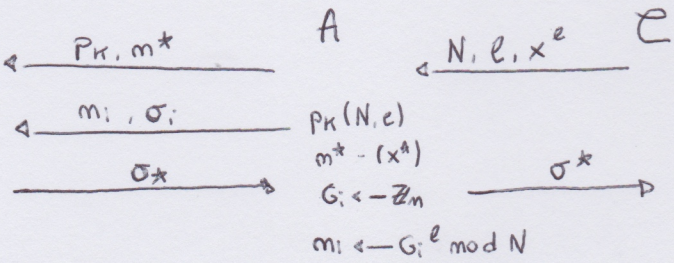
$\sigma_i = Sign(sk, m_i)$

output 1 IFF $Verify(pk, m^*, \sigma^*)$

Definition Π is RUF-RMA secure if $\mathbb{P}[GAME^{RUF-RMA}_{\Pi,A}(\lambda) = 1] \leq negl(\lambda)$

b) THM - Under RSA Π is RUF-RMA secure

Proof - Reduction



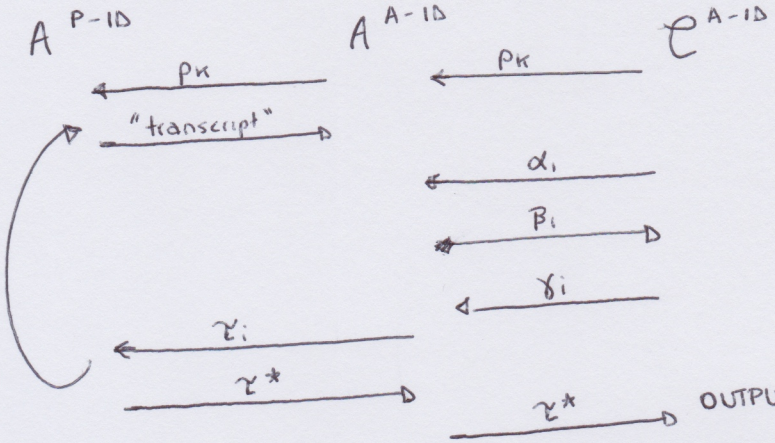
$(N, p, q, e, d) \leftarrow\!\$\ Gen$

$x \leftarrow\!\$\ \mathbb{Z}_m$

it breaks RSA assumption because $\sigma^* = m^{xd} \bmod N = x^{ed} \bmod N = x \bmod N$

EXERCISE 5

a) Active $\Rightarrow$ Passive

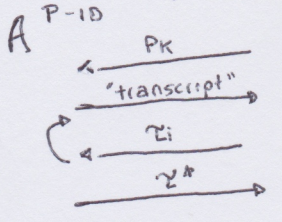


We can construct a Π_{BAD} s.t PASSIVE $\Rightarrow$ ACTIVE, where if we play as dishonest Verifier we can leak the entire sk

~~Traitor game~~

OUTPUT 1 IFF $V(pk, z^*) = 1$

For the passive game:

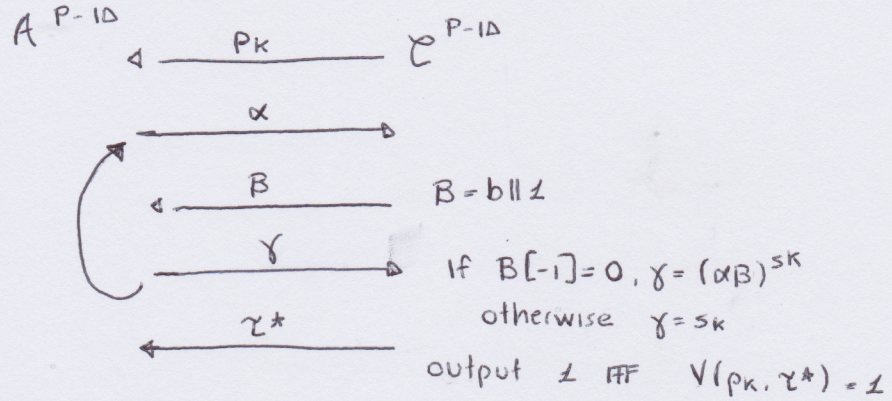


output 1 IFF $V(pk, z^*) = 1$

Now z_i will be as follows, from the point of view of A:

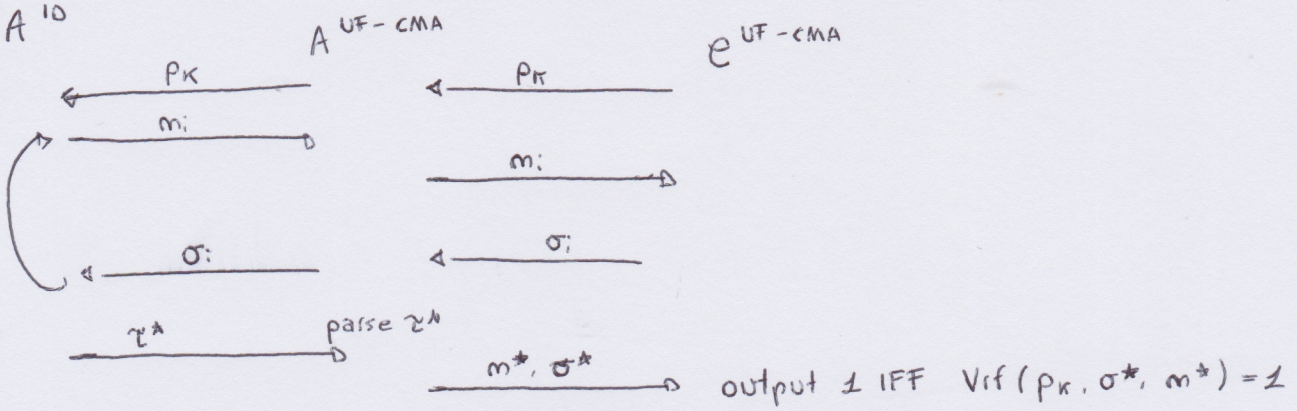
- $\alpha \leftarrow U$
- $B = b110$ where $b \leftarrow\!\$\ U$
- $\gamma = B[-1](sk) + (1 + B[-1])(\alpha\beta)^{sk}$

So the above ID-scheme has still Passive security. Instead for the active security we can play the following interaction:



Now that we have sk we can always create a valid z^*.

b) Suppose $\exists A^{ID}$ which is able to break the active security of π.



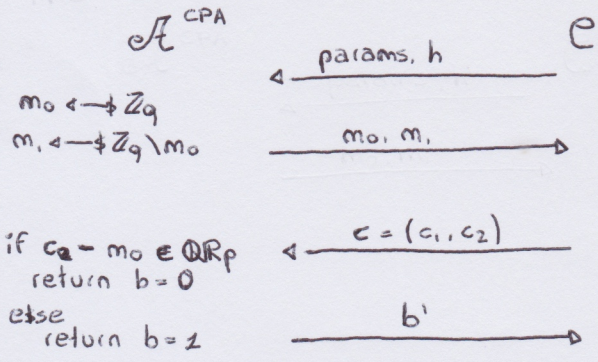
c) The HVZK property comes from the fact that the signature scheme used in the following way:

- the Verifier sends the message
- the Prover signs the message

doesn't leak any information about the secret used for signing the message (assuming V always as honest verifier). For the property of the signature it will never reveal anything about the secret used to generate σ.

EXERCISE 3c

The scheme is not CPA-secure.



The adversary acts as in this scheme, and the only case where he makes a mistake is when $c_2 - m_0 \in QR_p \mid b = 1$,

this case happens with $P \leq q/(2q+1)$. So:

$$P[A \text{ "wins"}] \geq \frac{1}{2} + \frac{1}{2} \cdot \frac{q+1}{2q+1}$$

Cryptography—Homework 2

Sapienza University of Rome
Master's Degree in Computer Science
Master's Degree in Cybersecurity
Master's Degree in Mathematics

Daniele Venturi

Due Date: December 18, 2019

1 Hashing — 25 Points

(a) Let $\mathcal{H} = \{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be a family of collision-resistant hash functions compressing $2n$ bits into n bits. Answer the following questions.

(i) Show that $\mathcal{H}$ is a seeded one-way function in the following sense: For all PPT adversaries A there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\Pr\Big[H_s(x') = y :\ s \leftarrow \{0,1\}^\lambda; x \leftarrow \{0,1\}^{2n}; y = H_s(x); x' \leftarrow \mathsf{A}(s,y)\Big] \leq \nu(n).$$

(ii) What happens in case the set of functions $\mathcal{H}$ is not compressing (i.e., the domain of each function H_s is also $\{0,1\}^n$)? Does collision resistance imply one-wayness in this case?

(b) Let $\mathcal{H} = \{H_s : \{0,1\}^{4n} \to \{0,1\}^{2n}\}_{s\in\{0,1\}^\lambda}$ and $\mathcal{H}' = \{H'_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^\lambda}$ be families of collision-resistant hash functions. Analyse the following candidate hash function family compressing $4n$ bits into n bits: $\mathcal{H}^* := \{H^*_{s_1,s_2}\{0,1\}^{4n} \to \{0,1\}^n\}_{s_1,s_2\in\{0,1\}^\lambda}$ such that $H^*_{s_1,s_2}(x) = H'_{s_2}(H_{s_1}(x))$ for $s_1, s_2 \leftarrow_\$ \{0,1\}^\lambda$.

2 Number Theory — 25 Points

(a) Recall that the CDH problem asks to compute g^{ab} given (g, g^a, g^b) for $(\mathbb{G}, g, q) \leftarrow_\$ \mathsf{GroupGen}(1^\lambda)$ and $a, b \leftarrow_\$ \mathbb{Z}_q$. Prove that the CDH problem is equivalent to the following problem: Given (g, g^a) compute g^{a^2}, where $(\mathbb{G}, g, q) \leftarrow_\$ \mathsf{GroupGen}(1^\lambda)$ and $a \leftarrow_\$ \mathbb{Z}_q$.

(b) Let $f_{g,p} : \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$ be the function defined by $f_{g,p}(x) := g^x \bmod p$. Under what assumption is $f_{g,p}$ one-way? Prove that the predicate $h(x)$ that returns the least significant bit of x is not hard-core for $f_{g,p}$.

(c) Let N be the product of 5 distinct odd primes. If $y \in \mathbb{Z}_N^*$ is a quadratic residue, how many solutions are there to the equation $x^2 = y \bmod N$?

3 Public-Key Encryption — 30 Points

(a) Show that for any CPA-secure public-key encryption scheme for single-bit messages, the length of the ciphertext must be super-logarithmic in the security parameter.

(b) Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\{0,1\}$ (i.e., for encrypting a single bit). Consider the following natural construction of a multi-bit PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with message space $\{0,1\}^t$, for some polynomial $t = t(\lambda)$: (i) The key generation stays the same, i.e. $\mathsf{KGen}'(1^\lambda) = \mathsf{KGen}(1^\lambda)$; (ii) Upon input $m = (m[1], \ldots, m[t]) \in \{0,1\}^t$ the encryption algorithm $\mathsf{Enc}'(pk, m)$ outputs a ciphertext $c = (c_1, \ldots, c_t)$ where $c_i \leftarrow\!\!\$\ \mathsf{Enc}(pk, m[i])$ for all $i \in [t]$; (iii) Upon input a ciphertext $c = (c_1, \ldots, c_t)$ the decryption algorithm $\mathsf{Dec}'(sk, c)$ outputs the same as $(\mathsf{Dec}(sk, c_1), \ldots, \mathsf{Dec}(sk, c_t))$.

(i) Show that if Π is CCA1 secure, so is Π'.

(ii) Show that, even if Π is CCA2 secure, Π' is not CCA2 secure.

(c) Consider the following variant of El Gamal encryption. Let $p = 2q + 1$, let $\mathbb{G}$ be the group of squares modulo p (so $\mathbb{G}$ is a subgroup of $\mathbb{Z}_p^*$ of order q), and let g be a generator of $\mathbb{G}$. The private key is $(\mathbb{G}, g, q, x)$ and the public key is $(\mathbb{G}, g, q, h)$, where $h = g^x$ and $x \in \mathbb{Z}_q$ is chosen uniformly. To encrypt a message $m \in \mathbb{Z}_q$, choose a uniform $r \in \mathbb{Z}_q$, compute $c_1 := g^r \bmod p$ and $c_2 := h^r + m \bmod p$, and let the ciphertext be (c_1, c_2). Is this scheme CPA-secure? Prove your answer.

4 Signature Schemes — 20 Points

(a) Consider a weaker variant of UF-CMA in which the attacker receives (pk, m^*) at the beginning of the experiment, where the message m^* is uniformly random over $\mathbb{Z}_N^*$, and thus it has to forge on m^* after possibly seeing polynomially-many signatures σ_i on uniformly random messages $m_i \leftarrow\!\!\$\ \mathbb{Z}_N^*$ chosen by the challenger. Call this notion random-message unforgeability under random-message attacks (RUF-RMA).

Formalize the above security notion, and prove that UF-CMA implies RUF-RMA but not viceversa.

(b) Recall the textbook-version of RSA signatures.

$\mathsf{KGen}(1^\lambda)$**:** Run $(N, e, d) \leftarrow\!\!\$\ \mathsf{GenModulus}(1^\lambda)$, and let $pk = (e, N)$ and $sk = (N, d)$.

$\mathsf{Sign}(sk, m)$**:** Output $\sigma = m^d \bmod N$.

$\mathsf{Vrfy}(pk, m, \sigma)$**:** Output 1 if and only if $\sigma^e \equiv m \bmod N$.

Prove that the above signature scheme $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfies RUF-RMA under the RSA assumption.

5 Actively Secure ID Schemes — 30 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{P}, \mathsf{V})$ be an ID scheme. Informally, an ID scheme is actively secure if no efficient adversary A (given just the public key pk) can make V accept, even after A participates maliciously in poly-many interactions with P (where the prover is given both the public key pk and the secret key sk). More formally, we say that Π satisfies active security if for all PPT adversaries A there is a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that

$$\Pr\left[\mathbf{Game}^{\mathsf{mal\text{-}id}}_{\Pi,\mathsf{A}}(\lambda) = 1\right] \leq \nu(\lambda),$$

where the game $\mathbf{Game}^{\mathsf{mal\text{-}id}}_{\Pi,\mathsf{A}}(\lambda)$ is defined as follows:

- The challenger runs $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$, and returns pk to A.
- Let $q(\lambda) \in \mathrm{poly}(\lambda)$ be a polynomial. For each $i \in [q]$, the adversary can run the protocol Π with the challenger (where the challenger plays the prover and the adversary plays the malicious verifier), obtaining transcripts $\tau_i \leftarrow\!\!\$\ (\mathsf{P}(pk, sk) \rightleftarrows \mathsf{A}(pk))$.
- Finally, the adversary tries to impersonate the prover in an execution of the protocol with the challenger (where now the challenger plays the honest verifier), yielding a transcript $\tau^* \leftarrow\!\!\$\ (\mathsf{A}(pk) \rightleftarrows \mathsf{V}(pk))$.
- The game outputs 1 if and only if the transcript τ^* is accepting, i.e. $\mathsf{V}(pk, \tau^*) = 1$.

Answer the following questions.

(a) Prove that passive security is strictly weaker than active security. Namely, show that every ID scheme Π that is actively secure is also passively secure, whereas there exists a (possibly contrived) ID scheme Π_{bad} that is passively secure but not actively secure.

(b) Let $\Pi' = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme, with message space $\mathcal{M}$. Prove that if Π' is UF-CMA, the following ID scheme $\Pi = (\mathsf{KGen}, \mathsf{P}, \mathsf{V})$ (based on Π') achieves active security:

$\mathsf{P}(pk, sk) \rightleftarrows \mathsf{V}(pk)$**:** The verifier picks random $m \leftarrow\!\!\$\ \mathcal{M}$, and forwards m to the prover. The prover replies with $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk, m)$, and finally the verifier accepts if and only if $\mathsf{Vrfy}(pk, m, \sigma) = 1$.

(c) Is the above protocol honest-verifier zero-knowledge? Prove your answer.

Cryptography—Final Exam

Sapienza University of Rome
Master Degree in Cybersecurity
Master Degree in Computer Science
Master Degree in Mathematics

Daniele Venturi

February 20, 2018

1 PRGs with Weak Seeds — 10 Points

Let $G : \{0,1\}^m \to \{0,1\}^{2m}$ be a $(t_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}})$-secure PRG. Explain how to safely use G in a setting where the seed S, instead of being uniform, is such that $\mathbb{H}_2(S) \geq m - d$. Assuming $m = 128$ and $d = 8$, show how to choose the parameters $t_{\mathsf{prg}}, \varepsilon_{\mathsf{prg}}$ in such a way that your construction achieves security 2^{-80} against all adversaries running in time at most 2^{20}.

2 Selective Unforgeability — 10 Points

Define a variant of universal unforgeability against chosen-message attacks (UF-CMA) for digital signatures, in which the adversary has to commit to the message $m^* \in \mathcal{M}$ on which he will forge a signature before seeing the public key (where $\mathcal{M}$ is the message space); note that in the definition the adversary should still be allowed to sign arbitrary messages. Name your notion SF-CMA (i.e., selective unforgeability against chosen-message attacks). Prove or disprove:

(a) UF-CMA $\Rightarrow$ SF-CMA.

(b) SF-CMA $\Rightarrow$ UF-CMA.

3 Actively Secure ID Schemes — 10 Points

Let $\Pi = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ be an ID scheme. Informally, an ID scheme is actively secure if no efficient adversary $\mathcal{A}$ (given just the public key pk) can make $\mathcal{V}$ accept, even after $\mathcal{A}$ participates maliciously in polynomially many interactions with $\mathcal{P}$ (given both the public

key pk and the secret key sk). More formally, we say that Π satisfies active security if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there is a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that for any polynomial $n := n(\lambda)$ the following holds:

$$\Pr\left[\mathsf{out}_{\mathcal{V}}(\mathcal{A}_2(pk, s_n) \rightleftarrows \mathcal{V}(pk)) = 1 : \begin{array}{c}(pk, sk) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda); s_0 := \varepsilon \\ (\forall i \in [n]) s_i \leftarrow (\mathcal{P}(pk, sk) \rightleftarrows \mathcal{A}_1(pk, s_{i-1}))\end{array}\right] \leq \nu(\lambda),$$

where s_0 is the empty string, $s_i \in \{0, 1\}^*$ is some arbitrary state information, and where the probability is taken over the random coin tosses of algorithms Gen, $\mathcal{A}$, and $\mathcal{P}$. Answer the following questions.

(a) Let $\Pi' = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, with message space $\mathcal{M}$. Prove that if Π' is CCA secure, the following ID scheme Π (based on Π') achives active security:

$\mathsf{Gen}(1^\lambda)$**:** Run $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$ and output (pk, sk).

$\mathcal{P}(pk, sk) \rightleftarrows \mathcal{V}(pk)$**:** The verifier picks random $m \leftarrow\!\!\$\ \mathcal{M}$, and forwards $c \leftarrow\!\!\$\ \mathsf{Enc}(pk, m)$ to the prover. The prover replies with $m' = \mathsf{Dec}(sk, c)$, and finally the verifier accepts if and only if $m' = m$.

(b) Is the above protocol honest-verifier zero-knowledge? Prove your answer.

Cryptography—Final Exam

Sapienza University of Rome
Master Degree in Cybersecurity
Master Degree in Computer Science
Master Degree in Mathematics

Daniele Venturi

June 8, 2018

1 Seeded Extraction — 10 Points

Say that X is a (k, n)-source if $X \in \{0,1\}^n$, and the min-entropy of X is at least k. Explain how we can extract ℓ bits of uniform randomness from any (k, n)-source without relying on any computational assumption, and why a uniform seed is needed for this purpose. Then answer the following questions:

(a) Suppose that $\ell = 128$; what is the minimal amount of min-entropy needed in order to obtain statistical error $\varepsilon = 2^{-80}$? What is the entropy loss?

(b) Suppose that $k = 238$; what is the maximal amount of uniform randomness that you can obtain with statistical error $\varepsilon = 2^{-80}$? Explain how to obtain $\ell = 320$ using computational assumptions.

2 CBC Mode and CCA Security — 10 Points

Recall the CBC mode of operation: Given a message $m = (m_1, \ldots, m_t)$ consisting of t blocks $m_i \in \{0,1\}^n$, and a random key $\kappa \in \{0,1\}^\lambda$, the ciphertext is $c = (c_0, c_1, \ldots, c_n)$ where $c_0 \leftarrow\!\!\$\, \{0,1\}^n$, $c_i = P_\kappa(c_{i-1} \oplus m_i)$ for all $i \in [n]$, and $P : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$ is a secure pseudorandom permutation.

In class, we proved that CBC mode yields a CPA-secure secret-key encryption scheme. Show that CBC mode is *not* CCA secure.

3 Strong Unforgeability — 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. Answer the following questions.[1]

(a) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is allowed to forge even on messages m asked to the signing oracle, as long as the forged signature σ^* is fresh, i.e. $\sigma^* \neq \sigma$ where σ is the signature returned by the oracle. Call the latter notion, *strong* UF-CMA.

(b) Prove or disprove: Strong UF-CMA implies UF-CMA.

(c) Prove or disprove: UF-CMA implies strong UF-CMA.

[1] For each question asking to prove/disprove an implication between two notions, if you think the implication holds you must show a reduction from one definition to the other. On the other hand, if you think the implication does not hold you must exhibit a scheme which satisfies one definition but not the other.

Cryptography—Final Exam

Sapienza University of Rome
Master Degree in Cybersecurity
Master Degree in Computer Science
Master Degree in Mathematics

Daniele Venturi

July 20, 2018

1 PRGs as OWFs — 10 Points

Let $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ be a PRG with one-bit stretch. Prove that G is by itself a one-way function.

2 OFB Mode and CCA Security — 10 Points

Recall the OFB mode of operation: Given a message $m = (m_1, \ldots, m_t)$ consisting of t blocks $m_i \in \{0,1\}^n$, and a random key $\kappa \in \{0,1\}^\lambda$, the ciphertext is $c = (r_0, c_1, \ldots, c_t)$ where $r_0 \leftarrow\!\!\$\ \{0,1\}^n$, $r_i = F_\kappa(r_{i-1})$, and $c_i = r_i \oplus m_i$ for all $i \in [t]$, and $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$ is a secure pseudorandom function.

In class, we mentioned that OFB mode yields a CPA-secure secret-key encryption scheme. Show that OFB mode is *not* CCA secure.

3 Non-Adaptive and Weak Unforgeability — 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. Answer the following questions. (For each question asking to prove/disprove an implication between two notions, if you think the implication holds you must show a reduction from one definition to the other; on the other hand, if you think the implication does not hold, you must exhibit a scheme which satisfies one definition but not the other.)

(a) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is given (together with the public key) $q \in \mathsf{poly}(\lambda)$ message/signature pairs $(m_i, \sigma_i)_{i\in[q]}$, where the messages $(m_i)_{i\in[q]}$ are chosen by the

adversary non-adaptively (i.e., all the same time) before obtaining the public key. As in UF-CMA, in order to win the game, the adversary then needs to forge on a message m^* which is fresh (i.e., not equal to any of the messages $m_1, \ldots, m_q$). Call the latter notion UF-naCMA.

(b) Formally define a variant of universal unforgeability against chosen-message attacks (UF-CMA), where the adversary is given (together with the public key) $q \in \mathsf{poly}(\lambda)$ message/signature pairs $(m_i, \sigma_i)_{i \in [q]}$, where each of the messages $(m_i)_{i \in [q]}$ is drawn uniformly at random from the message space. As in UF-CMA, in order to win the game, the adversary then needs to forge on a message m^* which is fresh (i.e., not equal to any of the messages $m_1, \ldots, m_q$). Call the latter notion UF-RMA.

(c) Prove or disprove: UF-naCMA implies UF-RMA.

(d) Prove or disprove: UF-naCMA implies UF-CMA.

Cryptography—Final Exam

Sapienza University of Rome
Master Degree in Cybersecurity
Master Degree in Computer Science
Master Degree in Mathematics

Daniele Venturi

January 24, 2019

1 MACs Combiner — 10 Points

Let $\Pi_1 = \mathsf{Tag}_1$, $\Pi_2 = \mathsf{Tag}_2$, and $\Pi_3 = \mathsf{Tag}_3$ be three deterministic MACs over key space $\mathcal{K} = \{0,1\}^\lambda$. You know that at least one of Π_1, Π_2, Π_3 is UF-CMA, but you don't know which one. Show how to design a secure MAC $\Pi^* = \mathsf{Tag}^*$ over key space $\mathcal{K}^3$ by combining Π_1, Π_2, and Π_3. Formally prove security of your candidate construction.

2 Replayable Security — 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption (PKE) scheme. Recall the game $\mathbf{G}^{\mathsf{pke\text{-}cca}}_{\Pi,\mathcal{A}}(\lambda, b)$ defining CCA security, involving an adversary $\mathcal{A}$:

1. The challenger picks $(pk, sk) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$, and forwards pk to $\mathcal{A}$.
2. $\mathcal{A}$ can ask polynomially many decryption queries. Upon input a query c, the challenger returns $m = \mathsf{Dec}(sk, c)$ to $\mathcal{A}$.
3. $\mathcal{A}$ chooses two messages m_0, m_1 of equal length, and receives $c^* \leftarrow\!\!\$\ \mathsf{Enc}(pk, m_b)$.
4. $\mathcal{A}$ can keep asking decryption queries, provided that these queries are different from the challenge ciphertext c^* defined in step 3.
5. $\mathcal{A}$ outputs a bit b'.

We say that a PKE scheme Π is CCA-secure if $\{\mathbf{G}^{\mathsf{pke\text{-}cca}}_{\Pi,\mathcal{A}}(\lambda, 0)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{G}^{\mathsf{pke\text{-}cca}}_{\Pi,\mathcal{A}}(\lambda, 1)\}_{\lambda\in\mathbb{N}}$.

Consider now a modified game $\mathbf{G}^{\mathsf{pke\text{-}rcca}}_{\Pi,\mathcal{A}}(\lambda, b)$ that is identical to the above game, except that step 4. is modified as follows:

4'. $\mathcal{A}$ can ask polynomially many decryption queries. Upon input a query c, the challenger computes $m = \mathsf{Dec}(sk, c)$; if $m \in \{m_0, m_1\}$ return test to $\mathcal{A}$, and otherwise return m.

We say that Π is *replayable* CCA (RCCA) secure if $\{\mathbf{G}^{\text{pke-rcca}}_{\Pi,\mathcal{A}}(\lambda, 0)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{G}^{\text{pke-rcca}}_{\Pi,\mathcal{A}}(\lambda, 1)\}_{\lambda\in\mathbb{N}}$. Finally, recall that Π is CPA-secure if no decryption query is allowed at all (i.e., step 2. and 4./4'. in the above games are ignored). Prove or disprove:

(a) CPA security $\Rightarrow$ RCCA security.

(b) RCCA security $\Rightarrow$ CPA security.

(c) RCCA security $\Rightarrow$ CCA security.

For each statement, if you think the property holds you need to give an explicit reduction from one notion to the other; on the contrary, if you think the property does not hold, you need to describe a PKE scheme that satisfies one notion but not the other notion.

3 +3-DDH — 10 Points

Let GroupGen be a PPT algorithm taking as input the security parameter, and outputting the description of a cyclic group $(\mathbb{G}, \cdot)$ of order a prime q, together with a generator $g \in \mathbb{G}$ of the group. Consider the following variant of the standard Decisional Diffie-Hellman (DDH) assumption, dubbed +3-DDH: We say that the +3-DDH assumption holds w.r.t. GroupGen if for all PPT adversaries $\mathcal{A}$ we have that

$$\left|\Pr\left[\mathcal{A}(\mathsf{params}, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{params}, g^x, g^y, g^{xy+3}) = 1\right]\right| \in \mathtt{negl}(\lambda),$$

where the probabilities are over the random coins of $\mathcal{A}$, and over the choice of $\mathsf{params} = (\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$, and $x, y \leftarrow\!\!\$\ \mathbb{Z}_q^*$. Prove that DDH implies +3-DDH.

Cryptography—Final Exam

Sapienza University of Rome
Master Degree in Cybersecurity
Master Degree in Computer Science
Master Degree in Mathematics

Daniele Venturi

February 11, 2019

1 Hash Functions Combiners — 10 Points

Let $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ be three families of hash functions, where

$$\forall i \in [3]: \ \mathcal{H}_i = \{H_s^i : \{0,1\}^* \to \{0,1\}^{\ell_i}\}_{s \in \{0,1\}^\lambda}.$$

You know that at least one of $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ is collision resistant, but you don't know which one.

(a) Show how to design a collision-resistant family $\mathcal{H}^* = \{H_s^* : \{0,1\}^* \to \{0,1\}^{\ell^*}\}_{s \in \{0,1\}^{3\lambda}}$, for some $\ell^* \in \mathbb{N}$ to be determined, by combining $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$.

(b) Does your construction also work for pre-image resistance (i.e. assuming at least one of $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ is one-way, is $\mathcal{H}^*$ also one-way)?

2 Multi-Message CPA Security — 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme for variable-length messages. Consider a generalization of CPA security—dubbed *multi-message CPA security*—where the adversary (after seeing the public key) can specify two vectors of messages of size $q \in \mathsf{poly}(\lambda)$, say $\vec{m}_0 = (m_0^1, \ldots, m_0^q)$ and $\vec{m}_1 = (m_1^1, \ldots, m_1^q)$, where $|m_0^i| = |m_1^i|$ for all $i \in [q]$. The challenge ciphertext $\vec{c} = (c_1, \ldots, c_q)$ consists of the component-wise encryption of one of the two vectors.

Formalize the above notion using the indistinguishability paradigm. Hence, prove that standard CPA security implies multi-message CPA security.

3 Weak PRFs — 10 Points

A family of functions $\mathcal{F} = \{F_k : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ is called a weak pseudorandom function (wPRF) family if it is like a standard PRF family, except that in the security definition the distinguisher is not allowed to choose the evaluation points $x \in \mathcal{X}$, but rather those are chosen *uniformly at random* by the challenger, and later given to the distinguisher together with the corresponding outputs $y \in \mathcal{Y}$ (either real or random).

(a) Formalize the above definition and show that any PRF family is also a wPRF family.

(b) Show that there is a family $\mathcal{F}^*$ that is weakly pseudorandom but not pseudorandom.

(c) Let GroupGen be a PPT algorithm taking as input the security parameter, and outputting the description of a cyclic group $(\mathbb{G}, \cdot)$ of prime order q, together with a generator $g \in \mathbb{G}$ of the group. Consider the PRF family $\mathcal{F}_{\mathsf{params}} = \{F_k : \mathbb{G} \to \mathbb{G}\}_{k \in \mathbb{Z}_q}$, with hard-coded parameters $\mathsf{params} = (\mathbb{G}, g, q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$, and specified by

$$F_k(h) = h^k \in \mathbb{G}.$$

Prove that $\mathcal{F}$ is weakly pseudorandom under the DDH assumption.

CRYPTO_24-10-14

LAST LECTURE: $\mathcal{F}(\mathcal{H}) = \{F_K(h_s(\cdot))\}_{s,K \in \{0,1\}^n}$
is a PRF (and thus FIL MAC) for domain $\{0,1\}^N$ and $N \geq n$

Condition: $\mathcal{F}$ a PRF and $\mathcal{H}$ is AU. (almost universal)

New angle. Fine to have $\mathcal{H}$ COMPUTATIONAL AU
(using $\mathcal{F}'$)

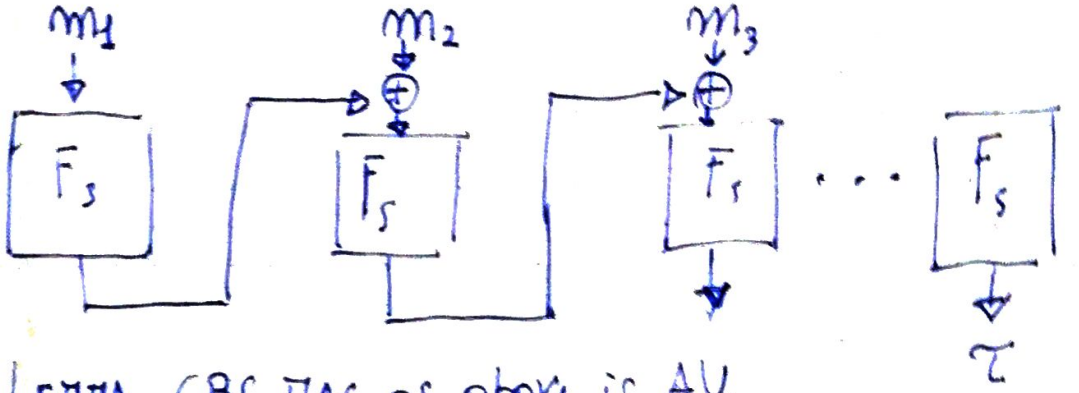
$$A \xrightarrow{x, x'} \mathcal{C}_{AU} \quad s \leftarrow \{0,1\}^n$$

$$\text{WIN}: x \neq x' \wedge h_s(x) = h_s(x')$$

Use two PRF keys K, s for $F_K(\cdot)$ and F_s
Trick. Just use K and separate domain $F_K(0\|\cdot)$, $F_K(1\|\cdot)$

CBS-MAC: $h_s(m_1, \dots, m_t) =$

$F_s \dots F_s(m_3 \oplus F_s(m_2 \oplus F_s(m_1)))$ $\Longrightarrow$ Actually already FIL MAC



LEMMA CBS-MAC as above is AU.

E-CBC$(m) = F_K(h_s(m))$

XOR-MAC Families F, H_0 $\quad m = (m_1, \ldots, m_t)$

Pick $\eta \leftarrow_S \{0,1\}^n$ $\quad \tau = (\eta, F_K(\eta) \oplus h_s(m))$ output

Which H_0? Note. given $m, \tau = (\eta, v)$ an attacker outputs

$m \neq m'$, $\tau' = (\eta, v \oplus a)$ for some a

Whenever $h_s(m) \oplus a = h_s(m')$ this is VALID

$$AU \Rightarrow a = 0$$
$$a \neq 0 \quad AXU$$

$h_s(m_1, \ldots, m_t) = F_s(m_1 \| 1) \oplus F_s(m_2 \| 2) \ldots \oplus F_s(m_t \| t)$

COMPUTATIONAL AXU

VIL? 1) AU approach does NOT work

$q_m(X) = m_1 + m_2 X + \ldots + m_t X^{t-1}$

CAN BE FIXED

2) CBC-MAC DOES NOT WORK

E-CBC WORKS.

3) XOR-MAC ✓

Attack on CBC-MAC for VIL.

$$\tau = F_K(\ldots F_K(m_2 \oplus F_K(m_1)) \ldots)$$

Attacker Ask m_1, get $\tau_1 = F_K(m_1)$

Let $m_2 = (m_1 \| m_1 \oplus \tau_1)$

Forgery: $m^* = m_2$; $\tau^* = \tau_1$

$Tag_K^{CBC}(m_2) = F_K(F_K(m_1) \oplus m_1 \oplus r_1) = r_1$

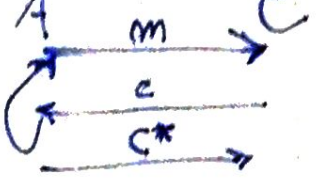
GOAL

Both: HIDE MESSAGE; CPA-security; CAN'T CHANGE MSG

$\Pi = (Enc, Dec)$

$Dec: \{0,1\}^\lambda \times \{0,1\}^m \to \{0,1\}^m \cup \{\perp\}$ ↳ INVALID

$GAME^{auth}_{\Pi,A}(\lambda)$



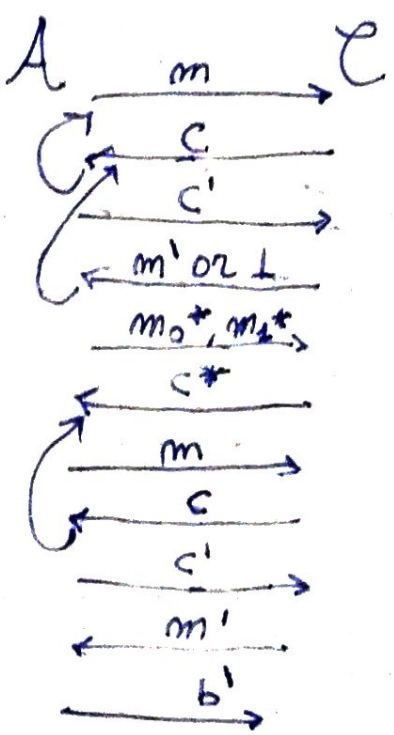
WIN

$c^* \neq \{c\}$

+ c^* VALID

CCA-SECURITY

$GAME^{cca}_{\Pi,A}(N,b)$



$K \leftarrow \{0,1\}^\lambda$

$c \leftarrow Enc(K,m)$, $c^* \leftarrow Enc(K, m_b^*)$

$m' = Dec(K, c')$

CONDITION

CANNOT ASK DECRYPTION of c^*!

$Enc(K, m) = (r, F_K(r) \oplus m)$ is CPA

EXERCISE Show not CCA

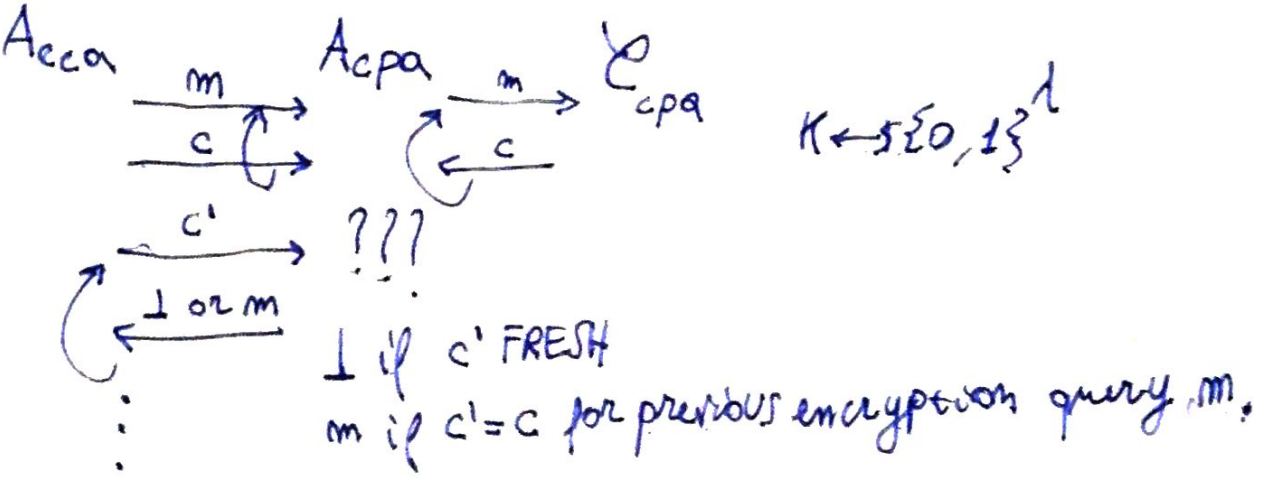
$Dec(K, (r, s)) = F_K(r) \oplus s$ $\quad r \Rightarrow r \quad m'$

Flip 1 bit of s $\Leftrightarrow$ $s \oplus 1 0 \ldots 0 = F_K(r) \oplus (m \oplus 1 0 \ldots 0)$

s' $\quad$ with the some r

THM CPA + AUTH $\Rightarrow$ CCA

Proof (SKetch) Given A_{cca} agains CCA, build A_{cpa} against CPA.



CONSTRUCT CCA — Given Π_{og} (UF-CMA) and (Enc, Dec) (CPA)

1) ENCRYPT-AND-MAC

$c \leftarrow\$ Enc(K, m)$; $\tau = Tog(K_2, m)$ $c' = (c, \tau)$

2) MAC-THEN-ENCRYPT

$\tau = Tog(K_2, m)$, $c \leftarrow\$ Enc(K_1, m \| \tau)$ $c' = (c, \tau)$

3) ENCRYPT-THEN-MAC $c \leftarrow\$ Enc(K_1, m)$ $c' = (c, \tau)$

$\tau = Tog(K_2, c)$

$Enc'(m)$ ← after writing

EXERCISE

What Works 1) NO

2) NO

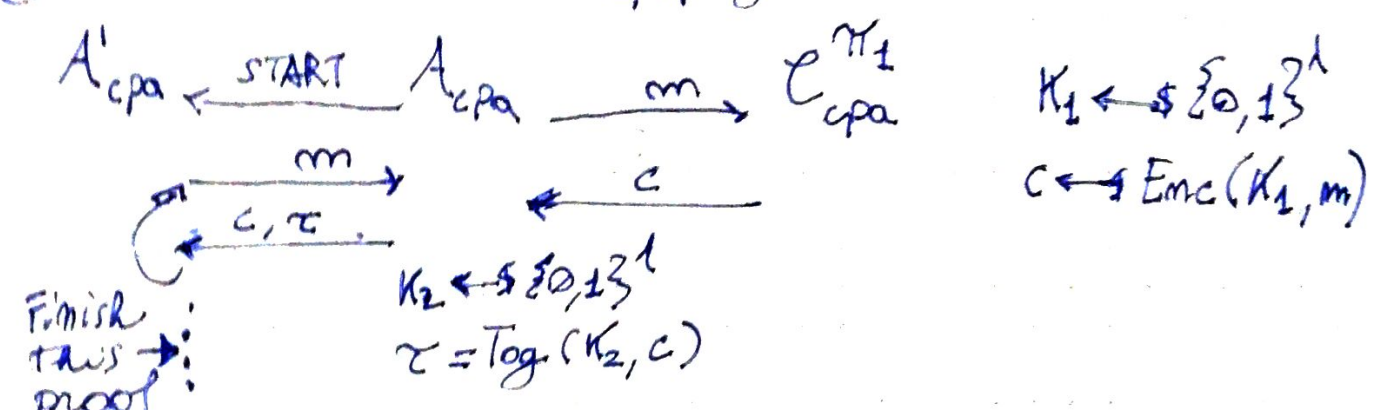
3) YES

THM If π_1 is CPA

π_2 is UF-CMA* $\Rightarrow \pi' = (Enc', Dec')$ CCA

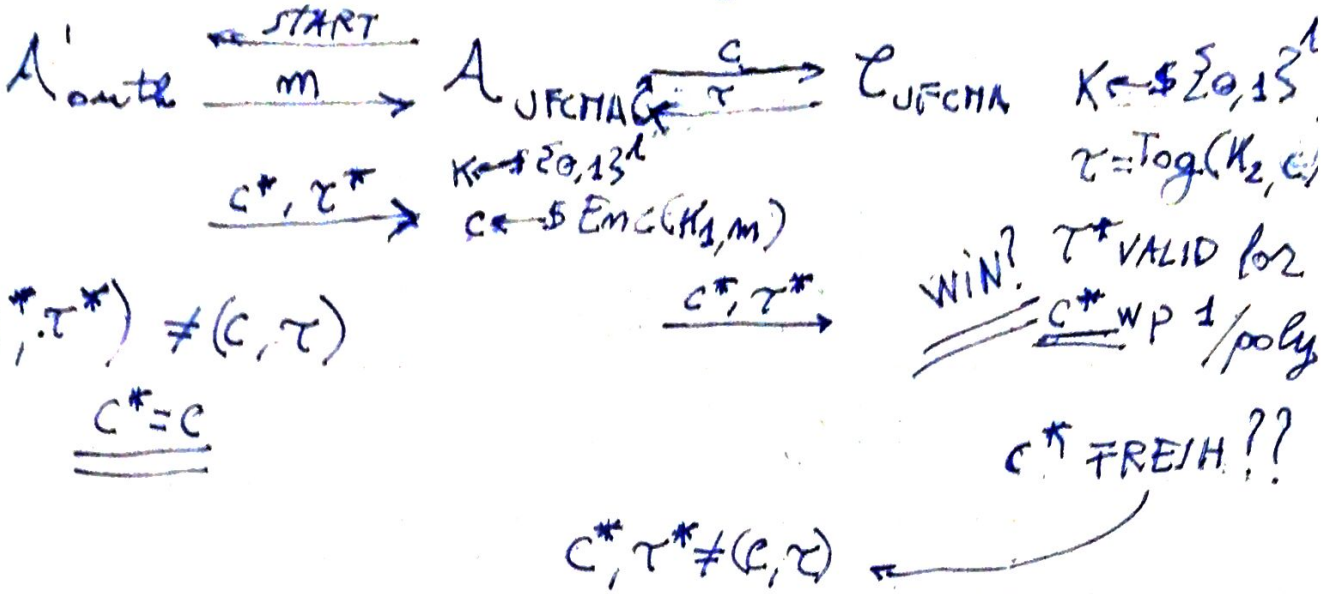
Proof. By THM suffices to prove AUTH + CPA (A) (B)

(B) Assume not $\exists$ PPT A'_{cpa} against π'^{CPA}



CONSTRUCT CCA

B) Assume not: ∃ PPT A'_{auth}



Cryptography—Final Exam

Sapienza University of Rome
Master's Degree in Cybersecurity
Master's Degree in Computer Science
Master's Degree in Mathematics

Daniele Venturi

May 5, 2020

1 A PRG Candidate — 10 Points

Let $f : \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a one-way permutation, and $G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+\ell}$ be a pseudorandom generator with positive stretch (i.e., $\ell \geq 1$). Analyze the following derived construction of a pseudorandom generator $G'_f : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+\ell}$, where

$$G'(s) := (f(s), G(s)).$$

In case you think the derived construction is not secure, exhibit a concrete attack; otherwise, provide a proof of security.

2 PKE Combiners — 10 Points

Let $\Pi_1 = (\mathsf{KGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ and $\Pi_2 = (\mathsf{KGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ be two PKE schemes with the same message space $\mathcal{M} = \{0,1\}^n$. You know that at least one of the two PKE schemes is secure, but you don't know which one. Show how to combine Π_1 and Π_2 into a PKE scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, with message space $\mathcal{M}$, such that Π satisfies CPA security as long as at least one of Π_1 and Π_2 satisfies CPA security.

3 ID Scheme based on RSA — 10 Points

Consider the following ID scheme $\Pi = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$.

- The key generation algorithm first computes parameters (N, e, d) as in the RSA cryptosystem. In particular, $N = p \cdot q$ for sufficiently large primes p, q, and $e \cdot d \equiv 1 \bmod (p-1)(q-1)$ with e a prime number. Hence, it picks $x \leftarrow\!\!\$\ \mathbb{Z}_N^*$, computes $y = x^e \bmod N$, and it returns $pk = (N, e, y)$ and $sk = x$.

- One execution of the ID scheme goes as follows: (1) The prover $\mathcal{P}$ picks $a \leftarrow\!\!\$\, \mathbb{Z}_N^*$, and sends $\alpha = a^e \bmod N$ to $\mathcal{V}$; (2) The verifier $\mathcal{V}$ forwards a random challenge $\beta \leftarrow\!\!\$\, \mathbb{Z}_e$ to $\mathcal{P}$; (3) The prover $\mathcal{P}$ replies with $\gamma = x^\beta \cdot a \bmod N$, where x is taken from the secret key and a is the same value sampled in the first round.
- The verifier $\mathcal{V}$ accepts a transcript $\tau = (\alpha, \beta, \gamma)$ if and only if

$$\gamma^e \cdot y^{-\beta} = \alpha \bmod N.$$

Prove that Π is a canonical ID scheme satisfying completeness, special soundness, and honest-verifier zero knowledge under the RSA assumption.

(**Hint:** To prove special soundness you can use the following fact: Given N, elements $u, v \in \mathbb{Z}_N^*$, and integers e, e' for which it holds that $\gcd(|e|, |e'|) = 1$ and $u^e = v^{e'} \bmod N$, an e-th root of v (modulo N) can be computed in polynomial time.)

Cryptography—Final Exam

Sapienza University of Rome
Master's Degree in Cybersecurity
Master's Degree in Computer Science
Master's Degree in Mathematics

Daniele Venturi

February 10, 2020

1 PRFs Combiners — 10 Points

For each $i \in [3]$, let $\mathcal{F}_i = \{F_k^i : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^\lambda}$ be a family of functions (i.e., one function for each choice of the secret key). You know that at least one of $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$ is a PRF family, but you don't know which one.

Show how to design a PRF family $\mathcal{F}^* = \{F_{k^*}^* \{0,1\}^n \to \{0,1\}^n\}_{k^* \in \{0,1\}^{3\lambda}}$ by combining $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$. (Note that the length of the secret key k^* is 3λ bits.)

2 Replayable CCA Security — 10 Points

Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption (PKE) scheme. Recall the game $\mathbf{G}^{\mathsf{cca}}_{\Pi,\mathcal{A}}(\lambda, b)$ defining chosen-ciphertext attacks (CCA) security, parameterized by Π, a PPT adversary $\mathcal{A}$, and hidden bit $b \in \{0,1\}$:

1. The challenger picks $(pk, sk) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$, and forwards pk to $\mathcal{A}$.
2. $\mathcal{A}$ can ask poly-many decryption queries. Upon input a query c, the challenger returns $m = \mathsf{Dec}(sk, c)$ to $\mathcal{A}$.
3. $\mathcal{A}$ chooses two messages m_0, m_1 of equal length, and receives $c^* \leftarrow_{\$} \mathsf{Enc}(pk, m_b)$.
4. $\mathcal{A}$ can keep asking decryption queries, provided that these queries are different from the challenge ciphertext c^*.
5. $\mathcal{A}$ outputs a bit b'.

We say that a PKE scheme Π is CCA-secure if $\{\mathbf{G}^{\mathrm{cca}}_{\Pi,\mathcal{A}}(\lambda,0)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{G}^{\mathrm{cca}}_{\Pi,\mathcal{A}}(\lambda,1)\}_{\lambda\in\mathbb{N}}$.

Consider now a modified game $\mathbf{G}^{\mathrm{rcca}}_{\Pi,\mathcal{A}}(\lambda,b)$ that is identical to the above game, except that step 4 is modified as follows:

4'. $\mathcal{A}$ can ask poly-many decryption queries. Upon input a query c, the challenger computes $m = \mathsf{Dec}(sk,c)$; if $m \in \{m_0,m_1\}$ return test to $\mathcal{A}$, and otherwise return m.

We say that Π is replayable CCA (RCCA) secure if $\{\mathbf{G}^{\mathrm{rcca}}_{\Pi,\mathcal{A}}(\lambda,0)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{G}^{\mathrm{rcca}}_{\Pi,\mathcal{A}}(\lambda,1)\}_{\lambda\in\mathbb{N}}$. Finally, recall that Π is CPA-secure if no decryption query is allowed at all (i.e., steps 2,4,4' in the above games are ignored). Prove or disprove:

(a) CPA security $\Rightarrow$ RCCA security.

(b) RCCA security $\Rightarrow$ CPA security.

(c) RCCA security $\Rightarrow$ CCA security.

For each statement, if you think the property holds you need to give an explicit reduction from one notion to the other; on the contrary, if you think the property does not hold, you need to describe a PKE scheme that satisfies one notion but not the other notion.

+1-DDH — 10 Points

Let GroupGen be a PPT algorithm taking as input the security parameter, and outputting the description of a cyclic group $(\mathbb{G},\cdot)$ of prime order q, together with a generator $g \in \mathbb{G}$ of the group. Consider the following variant of the standard Decisional Diffie-Hellman (DDH) assumption, dubbed +1-DDH: We say that the +1-DDH assumption holds w.r.t. GroupGen if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\Pr\left[\mathcal{A}(1^\lambda,\rho,g^x,g^y,g^{xy}) = 1\right] - \Pr\left[\mathcal{A}(1^\lambda,\rho,g^x,g^y,g^{xy+1}) = 1\right] \leq \nu(\lambda),$$

where the probabilities are over the random coins of $\mathcal{A}$, and over the choice of $\rho = (\mathbb{G},g,q) \leftarrow\!\!\$\ \mathsf{GroupGen}(1^\lambda)$, and $x,y \leftarrow\!\!\$\ \mathbb{Z}_q$. Prove that DDH implies +1-DDH.

Cryptography—Final Exam

Sapienza University of Rome
Master's Degree in Cybersecurity
Master's Degree in Computer Science
Master's Degree in Mathematics

Daniele Venturi

January 14, 2020

1 Hash Combiners

10 Points

Let $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ be three families of hash functions, where

$$\forall i \in [3] : \ \mathcal{H}_i = \{H^i_s : \{0,1\}^* \to \{0,1\}^{\ell_i}\}_{s \in \{0,1\}^\lambda}.$$

You know that at least one of $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ is collision resistant, but you don't know which one. Show how to design a collision-resistant family $\mathcal{H}^* = \{H^*_s : \{0,1\}^* \to \{0,1\}^{\ell^*}\}_{s \in \{0,1\}^{3\lambda}}$, for some $\ell^* \in \mathbb{N}$ to be determined, by combining $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$.

2 Weak PRFs

10 Points

A family of functions $\mathcal{F} = \{F_k : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$ is called a weak pseudorandom function (wPRF) family if it is like a standard PRF family, except that in the security definition the distinguisher is not allowed to choose the evaluation points $x \in \mathcal{X}$, but rather those are chosen *uniformly at random* by the challenger, and later given to the distinguisher together with the corresponding outputs $y \in \mathcal{Y}$ (either real or random).

(a) Formalize the above definition and show that any PRF family is also a wPRF family.

(b) Show that there is a family $\mathcal{F}^*$ that is weakly pseudorandom but not pseudorandom.

(c) Let $\mathsf{GroupGen}$ be a PPT algorithm taking as input the security parameter, and outputting the description of a cyclic group $(\mathbb{G}, \cdot)$ of prime order q, together with a generator $g \in \mathbb{G}$ of the group. Consider the PRF family $\mathcal{F}_{\mathsf{params}} = \{F_k : \mathbb{G} \to \mathbb{G}\}_{k \in \mathbb{Z}_q}$, with hard-coded parameters $\mathsf{params} = (\mathbb{G}, g, q) \leftarrow_{\$} \mathsf{GroupGen}(1^\lambda)$, and specified by

$$F_k(h) = h^k \in \mathbb{G}.$$

Prove that $\mathcal{F}$ is weakly pseudorandom under the DDH assumption.

3 Selective Unforgeability — 10 Points

A signature scheme $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is called selectively unforgeable against chosen-message attacks (SUF-CMA) if no efficient attacker can forge a signature on a message m^* that is chosen before seeing the public key, even if afterwards (i.e., after learning the public key) the adversary can observe signatures on messages $m \neq m^*$ of its choice.

Formalize the above notion, and show that it is equivalent to universal unforgeability against chosen-message attacks (UF-CMA) as long as the message space $\mathcal{M}$ satisfies $|\mathcal{M}| \in O(\log \lambda)$, where $\lambda \in \mathbb{N}$ is the security parameter.