Homework 2

solutions by Luigi Russo, 1699981

1 Hashing

(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 $\mathtt{A}$ there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that

$$\mathbb{P}[H_s(x') = y : s \leftarrow \{0,1\}^\lambda; x \leftarrow \{0,1\}^{2n}; y \leftarrow H_s(x); x' \leftarrow \mathtt{A}(s,y)] \leq \nu(n)$$

SOL

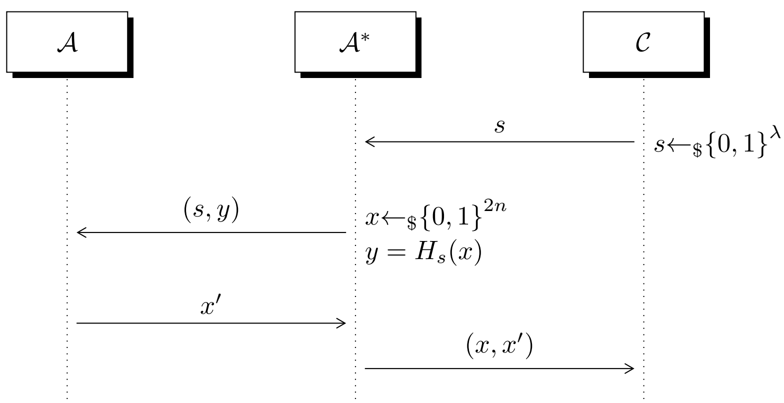

Figure 1: Base an attack to CR $\mathcal{H}$ family, assuming the existence of a PPT attacker $\mathcal{A}$, able to win the seeded one-way function Game with non negligible probability.

Assume there exists a PPT attacker $\mathcal{A}$ that given a value $y = H_s(x)$, with $s \leftarrow_{\$} \{0,1\}^\lambda$ and $x \leftarrow_{\$} \{0,1\}^{2n}$, finds a pre-image $x' : H_s(x) = H_s(x')$ with non negligible probability $\geq \frac{1}{p(n)}$ for some $p(n) \in poly(n)$. Then we can build on top of $\mathcal{A}$ an attacker $\mathcal{A}^*$ able to break the collision resistance of $\mathcal{H}$.

In particular, the challenger selects and fixes a seed s; $\mathcal{A}^*$ fixes a value x and forwards to $\mathcal{A}$ the value $y = H_s(x)$; $\mathcal{A}$ returns a value x' and the tuple (x, x') is forwarded to the challenger: the game is won if $x \neq x'$ and $H_s(x) = H_s(x')$. Now, by construction, we know that $\mathcal{A}$ "inverts"[1] y with probability $\geq \frac{1}{p(n)}$. We also know that the domain of $\mathcal{H}$ is the set of $2n-$bit strings, while its co-domain is the set of $n-$bit strings: this implies that the expected value of elements $x : H_s(x) = y$, once we fix y and s, is 2^n, thus implying

[1] with some abuse, we mean that it finds a valid preimage of y.

that the probability for the attacker $\mathcal{A}$ to output a value x' equal to x is only negligible $(= 2^{-n})$. These two considerations allow to conclude that $\mathcal{A}^*$ outputs a valid and colliding pair (x, x') with probability $\geq \frac{1}{p(n)}(1 - 2^{-n}) \in poly(n)$. $\square$

(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?

SOL Consider $H_s(x){:=}x \oplus \texttt{pad}(s)$, where $\texttt{pad} : \{0,1\}^\lambda \rightarrow \{0,1\}^n$ is public padding function[2]. Then it is easy to see that $\mathcal{H}$ is collision resistant, since there is no collision at all (for each s, $H_s(\cdot)$ defines a permutation). By the way, this family of functions is definitely not one way; it is easily invertible, since the seed s is public in the game, i.e. it is known to the adversary too: indeed, given $y = H_s(x) = \texttt{pad}(s) \oplus x$, the attacker outputs $\mathcal{A}(s, y) = y \oplus \texttt{pad}(s) = \texttt{pad}(s) \oplus \texttt{pad}(s) \oplus x = x$. $\square$

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

SOL

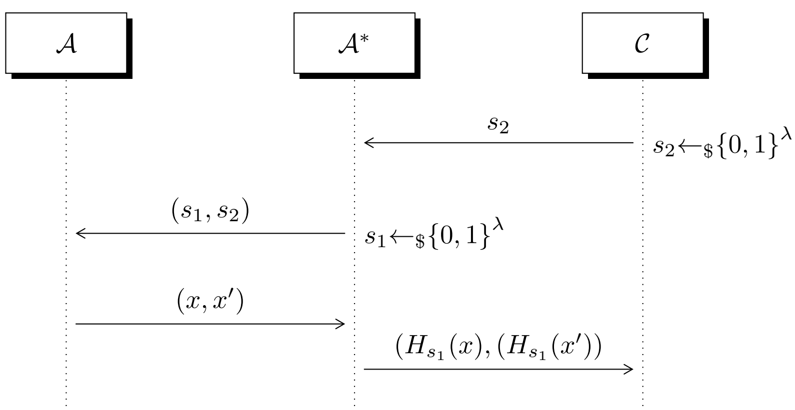

Figure 2: Base an attack to CR $\mathcal{H}'$ family on top of an attacker $\mathcal{A}$ to the collision resistance of $\mathcal{H}^*$.

Let assume that $\mathcal{H}^*$ is not collision resistant: this means that, by definition, there exists some PPT attacker $\mathcal{A}$ that, given two values as seed s_1, s_2, is able to find, with non negligible probability, a valid pair (x, x') such that $H^*_{s_1,s_2}(x) = H^*_{s_1,s_2}(x')$, with $x \neq x', x, x' \in \{0,1\}^{4n}$. If this is the case we can try to build an attacker $\mathcal{A}^*$ to break the collision resistance of $\mathcal{H}'$. Note that the challenger fixes a seed s_2 and waits for the attacker to find a collision; $\mathcal{A}^*$ randomly selects and fixes a seed $s_1 \in \{0,1\}^\lambda$ and uses $\mathcal{A}$ in order to find a pair (x, x'). By assumption, this pair is such that $H^*_{s_1,s_2}(x) = H^*_{s_1,s_2}(x')$ with non negligible probability $\geq \frac{1}{p(n)}$, for some $p(n) \in poly(n)$.

But if $\mathcal{A}$ is successful (i.e. $H^*_{s_1,s_2}(x) = H^*_{s_1,s_2}(x')$), either (i) $H_{s_1}(x) = H_{s_1}(x')$ or (ii) $H_{s_1}(x) \neq H_{s_1}(x')$. Case (i) can happen only with negligible probability $\epsilon(n)$, otherwise we could use $\mathcal{A}$ to build a PPT attacker to break the collision resistance of $\mathcal{H}$. Instead case (ii) guarantees that the pair is valid to build a successful attack to $\mathcal{H}'$: in both situations we come to an absurd given our initial assumptions. $\square$

[2] e.g. $\texttt{pad}(s){:=}s||0^{n-\lambda}$.

2 Number Theory

(a) Recall that the CDH problem asks to compute g^{ab} given (g, g^a, g^b) for $(G, g, q) \leftarrow_{\$} \texttt{GroupGen}(1^\lambda)$ 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 $(G, g, q) \leftarrow_{\$} \texttt{GroupGen}(1^\lambda)$ and $a \leftarrow_{\$} \mathbb{Z}_q$.

SOL For sake of simplicity, let us call SCDH the problem defined above (i.e. compute g^{a^2}, given a tuple $(\mathbb{G}, g, q, g^a)$). We have to prove that both CDH $\Rightarrow$ SCDH and SCDH $\Rightarrow$ CDH.

(CDH $\Rightarrow$ SCDH) given some g^a it is easy to see that this is equivalent to solve $\frac{\text{CDH}(g, g^a, g^{a+r})}{g^{ar}} = \frac{g^{aa+ar}}{g^{ar}} = g^{a^2}$, for a random choiche of $r \leftarrow_{\$} \mathbb{Z}_q$.

(SCDH $\Rightarrow$ CDH) given (g^a, g^b), we can compute $\frac{\text{SCDH}(g^{a+b})}{\text{SCDH}(g^a) \cdot \text{SCDH}(g^b)} = \frac{g^{a^2+2ab+b^2}}{g^{a^2} \cdot g^{b^2}} = g^{2ab}$: at this point we need to compute the square root of such a value: generally speaking this operation cannot be done efficiently; however, there are some groups where computing the square root can be done very easily (e.g. cyclic groups or when the order q is odd). For example, consider the case for q odd: it holds that $(g^{2ab})^{\frac{q+1}{2}} = 1 \cdot (g^{2ab})^{\frac{1}{2}} = g^{ab}$.

□

(b) Let $f_{g,p} : \mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^*$ 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 $f_{g,p}$.

SOL The function $f_{g,p}(\cdot)$ is one-way if g is a generator of Z_p^*, p is prime and under the DL assumption. First of all we know that this function is efficiently computable: this is a requirement for all OWF. On the contrary it is hard to "invert", i.e. to find a value $x' : f(x) = f(x')$. Indeed, given $y = g^x \bmod p$, a PPT attacker should find a value $x' : g^x \bmod p = g^{x'} \bmod p$. But this is possible if an only if $x - x' \equiv 0 \bmod (p-1)$.

Note that an element $g^x \in \mathbb{QR}_p$ if and only if the least significant bit of the exponent x is 0. Without loss of generality, indeed, we assume $x \in \{0,1\}^t$: this means that x can be written as $\sum_{i=0}^{t} x_i \cdot 2^i$. Given $f_{g,p}(x) = g^x \bmod p = g^{\sum_{i=0}^{t} x_i \cdot 2^i} \bmod p$, we can check whether $f_{g,p}(x) \in \mathbb{QR}_p$ or not, in order to leak the least significant bit of x.

The last step is to find a necessary and sufficient condition to check whether an element y is a quadratic residue modulo p or not.

We can compute $s = y^{\frac{p-1}{2}}$; if $s \equiv 1 \bmod p$, then y is a quadratic residue; otherwise it is not.

($\Rightarrow$) If y is a quadratic residue, then it holds that $y = (g^\alpha)^2 = g^{2\alpha}$, for some α. But then: $y^{\frac{p-1}{2}} = g^{\alpha(p-1)} = (g^{p-1})^\alpha = 1^\alpha = 1 \bmod p$.

($\Leftarrow$) If it holds that $y^{\frac{p-1}{2}} \equiv 1 \bmod p$, then $g^{\frac{x(p-1)}{2}} \equiv 1 \bmod p$. This implies that $\frac{x(p-1)}{2} = 0 \bmod (p-1)$ that can be satisfied if and only if x is even. Finally, we conclude that $y = g^x = (g^{\frac{x}{2}})^2$, proving that y is a quadratic residue.

□

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

SOL $x^2 \equiv y \bmod N$ means that $x^2 \equiv y \bmod p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5$. This implies that there exists a $k \in \mathbb{Z}$ such that $y = x^2 + k \prod_{i=1}^{5} p_i$. If we define $y_i{:=}y \bmod p_i$, we can rewrite the equation as $y_i \equiv x^2 \bmod p_i, i \in \{1, \ldots, 5\}$. This system of equation can be safely rewritten as $x^2 \equiv y_i \bmod p_i$, but now, for the Chinese remainder theorem, we know that once we fix the values y_i for all i, there exists a unique solution satisfying the system[3]. Note that each y_i is such that it is equivalent (modulo p_i) to x_i^2, where $x_i{:=}x \bmod y_i$. Each of the 5 equations has two distinct solutions $\pm\alpha_i$, since $\alpha_i \not\equiv -\alpha_i \bmod p_i$, because $p_i \neq 2, \forall i$. And this proves that there are 2^5 distinct systems, each of these with a unique solution (for CR theorem). This implies that the number of solutions to the original equation is $2^5 = 32$. □

3 Public-Key Encryption

(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.

SOL If the length of the ciphertext is not super-logarithmic, we know that the number of possible ciphertexts is at most λ^c for some constant c. The attacker can compute a ciphertext c_0 associated to bit 0, then waits for the challenge c_b^* and compares the two values: if they are equal it returns 0, otherwise returns 1. Now, if the challenger selects $b = 0$, the attacker wins with probability at least λ^{-c}, otherwise wins with probability 1. Taking the weighted average we have a winning probability $w \geq \frac{1}{2} + \frac{1}{2\lambda^c}$.

But then $|w - \frac{1}{2}| \geq \frac{1}{2\lambda^c}$, where $2\lambda^c \in \mathit{poly}(\lambda)$. □

(b) Let $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ 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 $\Pi' = (\mathtt{KGen}', \mathtt{Enc}', \mathtt{Dec}')$ with message space $\{0,1\}^t$, for some polynomial $t = t(\lambda)$: (i) The key generation stays the same, i.e. $\mathtt{KGen}'(1^\lambda) = \mathtt{KGen}(1^\lambda)$; (ii) Upon input $m = (m[1], \ldots, m[t]) \in \{0,1\}^t$ the encryption algorithm $\mathtt{Enc}'(pk, m)$ outputs a ciphertext $c = (c_1, \ldots, c_t)$ where $c_i \leftarrow_{\$} \mathtt{Enc}(pk, m[i])$ for all $i \in [t]$; (iii) Upon input a ciphertext $c = (c_1, \ldots, c_t)$ the decryption algorithm $\mathtt{Dec}'(sk, c)$ outputs the same as $(\mathtt{Dec}(sk, c_1), \ldots, \mathtt{Dec}(sk, c_t))$.

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

SOL In order to prove the above claim, it is convenient to use the hybrid argument: note that the goal is to prove that for all PPT adversaries $\mathcal{A}$, it holds that $\mathtt{Game}^{\mathtt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 0) \approx_c \mathtt{Game}^{\mathtt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 1)$.

In particular let us define a set of intermediate hybrid games $\mathtt{Hyb}_i$, where the adversary is given access to the decryption oracle (together with all the public parameters to make encryptions too, of course) exactly as each of the two games above: the difference is that, when the adversary commits the challenge tuple (m_0, m_1), the challenge ciphertext c^* is such that its j-th component is sampled from $\mathtt{Enc}(pk, m_0[j])$ if $i \leq j$, otherwise comes from $\leftarrow_{\$} \mathtt{Enc}(pk, m_1[j])$.

Note that once we fix the pair (m_0, m_1), the only difference between two hybrids $\mathtt{Hyb}_i$ and $\mathtt{Hyb}_{i+1}$ is in the distribution of the i-th component of the challenge c^*. It is easy to see that $\mathtt{Hyb}_1 \equiv \mathtt{Game}^{\mathtt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 0)$ and $\mathtt{Game}^{\mathtt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 1) \equiv \mathtt{Hyb}_t$.

Here for simplicity we give the distribution of the challenge ciphertext c^* for each hybrid game:

- $\mathtt{Hyb}_1 \rightarrow (\mathtt{Enc}(pk, m_0[1]), \mathtt{Enc}(pk, m_0[2]), \ldots, \mathtt{Enc}(pk, m_0[t])) \equiv \mathtt{Enc}'(m_0)$

[3] with a simple substitution, we put $T{:=}x^2$ and can apply the theorem.

- $\texttt{Hyb}_2 \to (\texttt{Enc}(pk, m_1[1]), \texttt{Enc}(pk, m_0[2]), \dots, \texttt{Enc}(pk, m_0[t]))$
- $\dots$
- $\texttt{Hyb}_{t-1} \to (\texttt{Enc}(pk, m_1[1]), \dots, \texttt{Enc}(pk, m_1[t-1]), \texttt{Enc}(pk, m_0[t]))$
- $\texttt{Hyb}_t \to (\texttt{Enc}(pk, m_1[1]), \dots, \texttt{Enc}(pk, m_1[t-1]), \texttt{Enc}(pk, m_1[t])) \equiv \texttt{Enc}'(m_1)$

Imagine there exists a PPT distinguisher between two hybrids $\texttt{Hyb}_i$ and $\texttt{Hyb}_{i+1}$ (for some i). If this is the case, we can build a PPT attacker $\mathcal{A}$ to break the CCA1-security of Π. Indeed, we can ask the challenger to give us an encryption c_b^* of a bit b. Then we pass to the distinguisher a value c such that the first i components are $\leftarrow_\$ \texttt{Enc}(pk, m_0[j]), \forall j < i$, its i-th component is equal to c_b^* and then the last $t-i-1$ components are $\leftarrow_\$ \texttt{Enc}(pk, m_1[j])$. We then forward to the challenger the same bit returned by the distinguisher and we retain the same non negligible advantage[4]. This proves that $\forall i, \texttt{Hyb}_i \approx_c \texttt{Hyb}_{i+1}$ which implies that also $\texttt{Game}^{\texttt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 0) \approx_c \texttt{Game}^{\texttt{CCA-1}}_{\Pi,\mathcal{A}}(\lambda, 1)$. $\square$

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

SOL Even if the original scheme is CCA2, it is very simple to break the CCA2-security of Π': the attacker obtains a ciphertext c_b^* where the first bit of m_0^* is 0 as well as the first bit of m_1^*. Now it produces another ciphertext for the bit 0: $x := \texttt{Enc}(0)$ such that x is different from the encryption of the first bit of m_b^*, and then asks for the decryption of $c' = (x || c_b^{**}$[5]$)$, and finally returns the decision bit $b' : m_{b'}^* = \texttt{Dec}(c')$, winning with probability 1. $\square$

(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.

SOL This scheme is not CPA-secure. Indeed, note that in order to compute c_1 we add an element $h^r \in \mathbb{G}$ to the message $m \in \mathbb{Z}_q$. This means that $c_2 \in \mathbb{Z}_p^*$ but not necessarily it is an element of $\mathbb{G}$: in particular, an attacker could exploit the fact that choosing uniformly a message $m \leftarrow_\$ \mathbb{Z}_q$, $\mathbb{P}[c_2 \in \mathbb{G} : c_2 = h^r + m \bmod p; h^r \in \mathbb{G}] = \frac{q}{2q+1}$, because among all the $2q+1$ elements of the group, only q of them are squares (and then belong to $\mathbb{G}$).

Note also that when $m = 0$, $c_2 = h^r + m = h^r \Rightarrow c_2 \in \mathbb{G}$.

So our attacker can use these two messages $m_0^* = 0$ and $m_1^* \leftarrow_\$ \mathbb{Z}_q$ and can receive the challenge $c_b^*(c_1^*, c_2^*)$: if $c_2^* \in \mathbb{G}$ it outputs 0, otherwise outputs 1. Note that if $b = 1$, the attacker wins with probability 1; otherwise it wins with probability $\frac{q+1}{2q+1}$. But this is enough to retain a non negligible advantage (the winning probability is $w = \frac{1}{2} + \frac{q+1}{2(2q+1)} \Rightarrow |w - \frac{1}{2}|$ is non negligible in λ). $\square$

4 Signature Schemes

(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 $\mathbb{Z}_{\mathbb{N}}^*$, and thus it has to

[4] this proof is not valid if $m_0[i] = m_1[i]$: but in such a scenario a distinguisher between $\texttt{Hyb}_i$ and $\texttt{Hyb}_{i+1}$ cannot even exist, since the i-th component is sampled from the same distribution and then the two distributions are equivalent.

[5] it is c_b^* without the first block, the one corresponding to the encryption of the first bit of m_b^*

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.

SOL Given a signature scheme $\Pi := (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Vrfy})$, let consider the following game $\mathtt{Game}_{\Pi,\mathcal{A}}^{\mathtt{RUF-RMA}}(\lambda)$:

(a) the challenger $\mathcal{C}$ generates $(pk, sk) \leftarrow_\$ \mathtt{KGen}(1^\lambda)$

(b) the challenger $\mathcal{C}$ fixes a message $m^* \leftarrow_\$ \mathbb{Z}_{\mathbb{N}}^*$ and forwards to $\mathcal{A}$ both the publick key pk and the message m^*

(c) $\mathcal{A}$ has access to oracle $O_1(sk)$: this oracle samples a message $m_i \leftarrow_\$ \mathbb{Z}_{\mathbb{N}}^*$ and returns (m_i, σ_i) such that $\mathtt{Vrfy}(pk, (m_i, \sigma_i)) = 1$

(d) $\mathcal{A}$ forges the signature σ^*

(e) the output of the game is $\mathtt{Vrfy}(pk, (m^*, \sigma^*))$ (i.e. 1 in case of win, 0 otherwise)

We say that a scheme Π is RUF-RMA-secure if $\forall$ PPT attackers $\mathcal{A}, \exists \epsilon(\lambda) \in negl(\lambda)$:

$$\mathbb{P}[\mathtt{Game}_{\Pi,\mathcal{A}}^{\mathtt{RUF-RMA}}(\lambda) = 1] \leq \epsilon(\lambda)$$

(UF-CMA $\Rightarrow$ RUF-RMA) Assume that a scheme $\Pi := (\mathtt{KGen}, \mathtt{Sign}, \mathtt{Vrfy})$ is not RUF-RMA: then there exists a PPT attacker $\mathcal{A}$ able to forge, with non negligible probability, a valid signature σ^* for the message m^* fixed by the challenger. If this is true, we can build on top of $\mathcal{A}$ an attacker $\mathcal{A}^*$ able to forge a valid pair and break UF-CMA-security too.

First of all, in this reduction, $\mathcal{A}^*$ forwards to $\mathcal{A}$ all the public parameters. Moreover, it randomly selects a message m^* and relays this to $\mathcal{A}$: m^* simulates the random challenge. At this point, the attacker $\mathcal{A}$ can start querying the oracle O_1; but $\mathcal{A}^*$ can simulate very easily these queries, simply picking a message at random $m_i \leftarrow_\$ \mathbb{Z}_{\mathbb{N}}^*$ and asking the challenger $\mathcal{C}$ to forge a valid signature σ_i: the pair (m_i, σ_i) is then forwarded to $\mathcal{A}$. After potentially a polynomial number of these oracle calls, $\mathcal{A}$ is ready to forge a signature σ^* and $\mathcal{A}^*$ forwards to $\mathcal{C}$ the pair (m^*, σ^*), retaining the same non-negligible advantage.

(RUF-RMA $\not\Rightarrow$ UF-CMA) The textbook version of RSA signatures is RUF-RMA (see next exercise), however it is not UF-CMA as proved in class. □

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

$\mathtt{KGen}(1^\lambda)$: Run $(N, e, d) \leftarrow_\$ \mathtt{GenModulus}(1^\lambda)$, and let $pk = (e, N)$ and $sk = (N, d)$.

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

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

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

SOL Assume not: then there exists a PPT attacker $\mathcal{A}$ (against RUF-RMA of Π), on top of which we can break RSA assumption. In particular, the challenger generates RSA params (N, e, d) and fixes, at random, a value $x \leftarrow_\$ \mathbb{Z}_{\mathbb{N}}^*$ and computes $y = x^e$. The attacker $\mathcal{A}^*$ forwards the public key (N, e) to $\mathcal{A}$; moreover it fixes $m^* = y$. In order to simulate the calls to oracle O_1 it can do the following: generates a signature $\sigma_i \leftarrow_\$ \mathbb{Z}_N^*$ and then computes the message $m_i = \sigma_i^e$ and returns the pair (m_i, σ_i). Note that such a pair is valid (because $\sigma_i^e = m_i$ by definition, and since σ_i is chosen at random, also m_i looks random[6]). We know that $\mathcal{A}$ forges a valid pair (y, σ^*) with non negligible probability, such that $\sigma^* : {\sigma^*}^e = y \Rightarrow \sigma^* = x$. This concludes the proof. □

5 Actively Secure ID Schemes

Let $\Pi = (\texttt{KGen}, \texttt{P}, \texttt{V})$ 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 $\texttt{V}$ accept, even after $\mathcal{A}$ participates maliciously in poly-many interactions with $\texttt{P}$ (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 $\mathcal{A}$ there is a negligible function $\nu : \mathbb{N} \in \{0, 1\}$ such that:

$$\mathbb{P}[\texttt{Game}^{\texttt{mal-id}}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \nu(\lambda),$$

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

- The challenger runs $(pk, sk) \leftarrow_\$ \texttt{KGen}(1^\lambda)$, and returns pk to $\mathcal{A}$.
- Let $q(\lambda) \in poly(\lambda)$ 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 $\tau_i \leftarrow_\$ (\texttt{P}(pk, sk) \rightleftarrows \mathcal{A}(pk))$.
- 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 $\tau^* \leftarrow_\$ (\mathcal{A}(pk) \rightleftarrows \texttt{V}(pk))$.
- The game outputs 1 if and only if the transcript τ^* is accepting, i.e. $\texttt{V}(pk, \tau^*) = 1$.

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.

SOL (active ⇒ passive) Assume not, then there exists some PPT attacker $\mathcal{A}$ that breaks the passive security of an actively secure scheme Π. If this is true, we can build a new PPT attacker $\mathcal{A}^*$ that breaks the active security of Π. Indeed, when $\mathcal{A}$ asks to see some valid transcript τ_i, prompting the empty string, $\mathcal{A}^*$ will simply start a session of the protocol as an **honest**[7] verifier with the challenger acting as the (honest) prover. The final transcript τ_i is then forwarded to the original attacker: and this can be done up to a polynomial number of times of course. At this point, $\mathcal{A}$ impersonates a malicious prover (because it has not the secret key sk), directly interacting with the verifier ($\mathcal{A}^*$, of course, forwards the messages of $\mathcal{A}$ to the challenger and then sends its replies back to $\mathcal{A}$). This allows $\mathcal{A}^*$ to retain the same non negligible advantage of $\mathcal{A}$ since the reduction is tight.

(passive ⇏ active) Consider a canonical Σ protocol Π: we define a variant Π' such that the verifier appends a bit b to its (unique) message β where $\mathbb{P}[b = 0] = 2^{-\lambda}$; if $b = 0$ (this happens only with negligible probability in λ) the prover has to send the secret key sk; otherwise it follows

[6] this is crucial, otherwise it would not be a valid simulation.
[7] must behave honestly, for each transcript requested by $\mathcal{A}$, in order to perfectly simulate the oracle.

the original protocol to compute γ[8]. This clearly does not compromise the passive security of Π', because the attacker should hope to see a transcript where the secret key is leaked, but this happens only with negligible probability since the verifier is honest (it is simulated by the challenger itself); on the contrary, this scheme is not actively secure, because the adversary is allowed to impersonate the verifier in the first part of its attack: then it can easily find out the secret key (needs only one query where it appends $b = 0$ to some β) and then it can act as a malicious prover and forge a valid transcript thanks to the leaked key. $\square$

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

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

SOL As always, assume not: then if there exists a PPT $\mathcal{A}$ able to break the active security of Π, we can build on top of it a new attacker $\mathcal{A}^*$ to break the UF-CMA-security of Π'. In our reduction, we first forward to the original attacker the public key pk (and, of course, all the public parameters generated by $\mathcal{C}$).

The attacker $\mathcal{A}$ is allowed in the first phase to interact as a (possibly malicious) verifier: it will start sending some message m_i, waiting for the prover to reply; $\mathcal{A}^*$ will forward m_i to the challenger in order to produce a valid σ_i and then is ready to reply; this allows the attacker $\mathcal{A}$ to collect some transcript τ_i. After polynomially many such queries, it tries to impersonate the prover: so $\mathcal{A}^*$ will pick a random (and "fresh"[9], i.e. $\neq m_i, \forall i$) message m^* and passes it to $\mathcal{A}$; then, a σ^* is forged (valid with probability $\geq \frac{1}{p(\lambda)}, p(\lambda) \in poly(\lambda)$); the pair (m^*, σ^*) is forwarded to the challenger and the winning condition is that σ^* is a valid signature of m^*. But this directly implies that $\mathcal{A}^*$ has the same non negligible advantage of the original $\mathcal{A}$. $\square$

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

SOL In order to be HVZK, it should exist a simulator S such that $\{pk, sk, \texttt{Trans}(pk, sk)\} \approx_c \{pk, sk, S(pk)\}$. Note that this simulator is given in input only the public key pk. But if such S exists, the underlying signature scheme Π cannot be UF-CMA: this because S is able to produce valid pairs (m_i, σ_i) in a way that is (computationally) indistinguishable from $\texttt{Trans}(pk, sk)$ and in particular it can forge a valid pair in $\texttt{Game}_{\Pi'}^{\texttt{UF-CMA}}(\lambda)$ too.

If, instead, we remove the UF-CMA assumption of Π', it could be that for some schemes it is possible to compute pairs (m_i, σ_i) given only pk: e.g. a possible strategy could be to pick a random σ_i and then find some message m_i such that $\texttt{Vrfy}(pk, (m_i, \sigma_i)) = 1$. If this is the case, then we can build such a simulator S and the above protocol is HVZK. $\square$

[8] of course $\texttt{V}'$ will check the last bit of β and if it is equal to 1, it will run $\texttt{V}(\cdot)$.

[9] note that it can do it, since it observes all the queries m_i.