Exam 2025A

Gidon Rosalki

2026-01-07

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_intro_to_crypto`

1 Question 1

Let there be a family of collision resistant functions $H_s \{0,1\}^{2n} \to \{0,1\}^n$. Let there also be a PRG $G : \{0,1\}^{n+1} \to \{0,1\}^{2n}$.

1.1 Part A

Consider

$$F_s(x_1 \| x_2) = H_s(H_s(x_1) \| H_s(x_2))$$

Where $|x_1|, |x_2| = 2n$, and $F_s : \{0,1\}^{4n} \to \{0,1\}^n$. Is the family F_s collision resistant?

Sol. Yes. Let us assume towards contradiction that F_s is not collision resistant, so there exists an adversary $\mathcal{A}$ that finds collisions in F_s. Let us create $\mathcal{B}(s)$, which runs $\mathcal{A}(s)$, which returns $(x_1, x_2), (x_1', x_2')$. Since $(x_1, x_2) \neq (x_1', x_2')$ then at least one of the pair of variables $x_i, x_i' : i \in \{1,2\}$ are different, so let us assume wlog that $x_1 \neq x_1'$. If so, then there are 2 cases:

1. $H_s(x_1) = H_s(x_1')$ in which case we are done, and return (x_1, x_1')
2. $H_s(x_1) \neq H_s(x_1')$ in which case we return $(H_s(x_1) \| H_s(x_2) \| H_s(x_1') \| H_s(x_2'))$

1.2 Part B

Consider

$$L_s(x) = H_s(G(x))$$

Where

$$L_s : \{0,1\}^{n+1} \to \{0,1\}^n$$

Is the family L_s collision resistant?

Sol. No. We will bring a counterexample of (H_s', G') such that L_s is not collision resistant.

$$H_s' = H_s \tag{1}$$

$$G'(x) = \begin{cases} G(x), & \text{if } x \notin \{0^n, 1^n\} \\ 0, & \text{if } x \in \{0^n, 1^n\} \end{cases} \tag{2}$$

Theorem 1 (Claim 1)**.** *G' is a PRG*

Proof . We will assume towards contradiction that there exists an adversary $\mathcal{A}$ that can differentiate between the output of G' and random, and from that build the adversary $\mathcal{B}$ that can differentiate between the output of G and random. It will be exactly the same adversary, and will have the same probability as G for differentiating between G and random, with the addition of $\frac{2}{2^n}$. Since finding this collision in G is in fact negligible, and the addition of $\frac{2}{2^n}$ is also negligible, then the finding of this collision is also in fact negligible. □

Theorem 2 (Claim 2)**.** *$L_s(x) = H_s(G'(x))$ is not a CRH*

Proof . Pretty trivial, since we know the definition of G', and may simply give L_s the two inputs such that G returns the same output, and we have found a non trivial collision in L_s □

2 Question 2

Let $f : \{0,1\}^n \to \{90,1\}^n$ be a one way function. We will use this to create a new signature scheme $\Pi = (Gen, \text{Sign}, \text{Vrfy})$:

- Gen $(1^n) = x_1, \ldots, x_n \leftarrow \{0,1\}^n$, $sk = (x_1, \ldots, x_n)$, $vk = (y_1, \ldots, y_n) = (f(x_1), \ldots, f(x_n))$
- Sign (sk, m): $\sigma = \perp$ (as in, empty string). For $1 \leq i \leq n$, if $m[j] = 1$, then $\sigma \| x_i$, then return σ
- Vrfy (vk, m): Passes over every bit in the message, and knows that the corresponding part of the message must be the preimage of a part of the vk, so it computes the function of it, and checks if it appears in the verification key

2.1 Part A

Show that the system is not secure as a one time signature.

Sol. It's trivial. The empty message 0^n will have the signature $\perp$, without even calling the oracle.

2.2 Part B

Correct the signature scheme such that it is now secure, and that the size of vk is $n^2 + n(\log(n) + 1)$ bits.

Sol. We will note that in the unaltered scheme, an adversary can change an arbitrary 1 in the message to a 0, by simply removing the relevant part of the signature. We can resolve this by signing the number of 0s in the message, which requires $\log(n) + 1$ bits, and thus the adversary cannot change the numbers of 0s, since he would also have to change the signature of the number of 0s:

- KeyGen (1^n): $sk = \left(x_1, \ldots, x_n, x^0_{n+1}, \ldots, x^0_{n+\log n+1}, x^1_{n+1}, \ldots, x^1_{n+\log n+1}\right)$
  $vk = \left(f(x_1), \ldots, f(x_n), f\left(x^0_{n+1}\right), \ldots, f\left(x^0_{n+\log n+1}\right), f\left(x^1_{n+1}\right), \ldots, f\left(x^1_{n+\log n+1}\right)\right)$
- Sign (sk, m): Sign $(m) \|$Lamport (zeroes (m))

This solves it in $n^2 + 2n(\log(n) + 1)$.

To prove it, let us assume towards contradiction that there exists adversary $\mathcal{A}$ that can win the game against this scheme. So, $\mathcal{A}$ outputs m, and receives in return from the oracle Sign (m), Lamport (zeroes (m)), and then at the end outputs m^*, sign (m^*), Lamport (zeroes (m^*)). There are now two cases:

1. Zeroes (m^*) = Zeroes (m): Then this message must be a permutation of another, since there are the same number of 0s. In this case, then we may break it similarly to how we did Lamport.
2. Zeroes $(m^*) \neq$ Zeroes (m): In this case, then we have succeeded, since we have created a new message with the same signature.

In order to remove the 2, then we may simply change KeyGen to remove the doubling of the bits from $x_{n+1}, \ldots, x_{n+\log n+1}$, and Sign to be Sign $(m \|$zeroes $(m))$. This may be proven with the exact same proof.

3 Question 3

3.1 Part A

Given a cyclic group (G, g, q) such that DDH holds $((g^x, g^y, g^{xy}) \approx (g^x, g^y, g^z))$, let there be two distributions:

$$\left(g^{a_1}, g^{a_2}, g^{a_1 b}, g^{a_2 b}\right) \tag{3}$$
$$\left(g^{a_1}, g^{a_2}, g^{r_1}, g^{r_2}\right) \tag{4}$$

Such that $a_1, a_2, r_1, r_2, b \leftarrow \mathbb{Z}_q$. Show that these distributions are indistinguishable.

Sol. Let us assume towards contradiction that they are distinguishable. So, we are building $\mathcal{A}(g^x, g^y, T)$ where $g \leftarrow \{g^{xy}, g^z\}$ that succeeds against DDH. We will do this by building $\mathcal{B}(g^x, g^{a_2}, T, (g^y)^{a_2})$. When T is random, then $\mathcal{B}$ has received lower option, and when T is g^{xy}, then $\mathcal{B}$ has received the top option. We have thus built an adversary that may win DDH.

3.2 Part B

We will define a key exchange protocol. In order for Alice and Bob to swap keys, Alice chooses $k, r \leftarrow \{0,1\}^n$, and sends Bob $s = k \oplus r$. Bob chooses $t \leftarrow \{0,1\}^n$, and sends $u = s \oplus t$. Alice sends Bob $w = u \oplus r$. Alice outputs k, and Bob outputs $w \oplus t$. Show that the protocol is correct, and whether or not it is secure.

Sol. Correctness:

$$\begin{aligned} w \oplus t &= u \oplus r \oplus t \\ &= s \oplus t \oplus r \oplus t \\ &= k \oplus r \oplus t \oplus r \oplus t \\ &= k \end{aligned}$$

Security: Not secure, in the slightest. The adversary observes $s = k \oplus r$, and $u = s \oplus t$. From this, they may compute $s \oplus u = k \oplus r \oplus k \oplus r \oplus t = t$. From there, like B, they have t, and when w is transmitted, they may compute $w \oplus t = k$, and find the secret key.