CS403 · Cryptography and Network Security

Assignment 4

Yubin Hu / 11712121

December 21, 2020

1

1.1

H_a is necessarily collison-resistant.

Here we use the contradiction method to prove. Suppose H_a is not necessarily collison-resistant, there is a PPT algorithm A which find a collison with non-negligible probability($x \neq y,\ H_a(x) = H_a(y)$). So we have $H_1^{s_1}(x)||H_2^{s_2}(x) = H_1^{s_1}(y)||H_2^{s_2}(y)$, and we got $H_1(x) = H_1(y)$ and $H_2(x) = H_2(y)$, which means H_1andH_2 are not collison-resistant. This conflicts with assumptions.

1.2

H_b is not necessarily collison-resistant.

Suppose we have H_1 be collison-resistant and $H_2(x) = 0$ for all x. So we got $\forall x, H_b(x) = H_1(0)||0$, and in this case, H_b is not necessarily collison-resistant.

1.3

H_c is necessarily collison-resistant.

H_c is necessarily collison-resistant. Suppose H_c is not necessarily collison-resistant, there is a PPT algorithm A which find a collison with non-negligible probability($x \neq y,\ H_c(x) = H_c(y)$). So we have $H_1(H_2(x)||x)||H_2(H_1(x)||x) = H_1(H_2(y)||y)||H_2(H_1(y)||y)$, and we got $H_1(H_2(x)||x) = H_1(H_2(y)||y)$ and $H_2(H_1(x)||x) = H_2(H_1(y)||y)$. Also we know that $x \neq y$, so $H_1(x)||x \neq H_1(y)||y$ and $H_2(x)||x \neq H_2(y)||y$, , which means H_1andH_2 are not collison-resistant. This conflicts with assumptions.

2

2.1

Suppose that we have the compression function $h : 0, 1^{2n} \rightarrow 0, 1^n$ and the length of k is n.

1. An arbitary message m with length n to ask the oracle. Let $t = Mac(m) = H(k||m) = h(h(k||IV)||m)$.

2. $Mac(m||t) = h()t||t$, since $Mac(m||t) = H(k||m||t) = h(h(h(k||IV)||m)||t)$ and $t = h(h(k||IV))$

Thus this PPT algorithm has a winning probability 1. This is not a secure MAC.

2.2

If H is modeled as a random oracle, $F_k(M) = H(k||m) is PRF$. And it is MAC security.

2.3

The random oracles are more soundness then the normal oracles. So the consequences are only available for the random oracles, not for all the cases.

3

3.1

There are the set of quadratic residue (QR) called G. $G \subseteq Z_n^*$. Suppose we have $y_1, y_2 \in G$, then $\exists x_1, x_2 \in Z_n^*$, $y_1 = x_1^2, y_2 = x_2^2$. So $y_1 y_2 = x_1^2 x_2^2 = x_1 x_2 x_1 x_2$, it is closure. And $1 = 1^2$, $1 \in G$, it is identity. Now we have $y \in G$, $\exists x, x^{-1} in G_n^*$, so that $y = x^2, xx^{-1} = 1$. $y^{-1} = x^{2^{-1}} = (x^{-1})^2$ and $y^{-1} \in G$, $yy^{-1} = 1$. Therefore, the set of QRs is a subgroup of Z_n^*.

3.2

- *if part* Suppose we have $y \in Z_p^*$, $log_g(y) = 2k$, then $g^{2k} = y, y = (g^k)^2$ and y is a QR.
- *only if part* Support that y is a QR, then $\exists x \in Z_p^*$ such that $y = x^2$. Let $log_g(x) = i$, then we got $x = g^i$ and $y = g^{2i}$.
- $log_g(y)$ is the min number over all the $2i\text{mod}(p-1)$.
- p is a prime, $p - 1 = 1$ or p is a even number. So $log_g(y)$ is even number.

4

Suppose we have Z_n has a generator g. $gcd(g, N) = 1$ and for $\forall y \in Z_N$. We know that in Z_N, $g^x = xg\text{mod}N$. Thus $x = g^{-1}y$ and We can obtain x through the extended Euclidean algorithm.

5

5.1

$S = 1, 4, 9, 16, 8, 2, 15, 13, 13, 15, 2, 8, 16, 9, 4, 1 = 1, 2, 4, 8, 9, 13, 15, 16$, the size is 8.

5.2

$x \in Z_{17}^*$ is a generator if and only if $gcd(x, \varphi(17)) = 1$. So the number of generator is $\varphi(\varphi(17)) = \varphi(16) = 8$

5.3

$g^{ab} \in S$, $ab\text{mod}2 = 0, 17$ is prime. $Pr[g^{ab} \in S] = 1 - Pr[ab\ is\ odd] = 1 - \frac{1}{4} = \frac{3}{4}$

6

Suppose x is a random element of Z_N^* and $y = x^2$, we have a algorithm A to get $y's$ square root z. if $z = \pm x$, we choose another x and do it again, until $z \neq \pm x$. So we got 5 square root of $y : \pm x \pm z$.

$a = x^2(modN) = z^2(modN)$, $x^2 - z^2 = (x+z)(x-z) = kN$, $x + z \neq 0$, $x - z \neq 0$, then $k \neq 0$ So $k = 1$ and $N = (x+z)(x-z)$

7

For arbitrary message m, we first choose an arbitrary $k \neq 0, 1$. Asking the signing oracle to sign $m' = mk^e$ mod N(here e is public key in the scheme of signature). Then we have the $Sign(m') = m^{fd}$ mod $N = (mk^e)^d$ mod $N = m^d \times k^{ed}$ mod $N = m^d$ mod N. Therefore, message m cannot be queried to the signing oracle.

8

Suppose G is a group with generator g, and h_1, h_2, h_3 are the element of G. $h_1 = g^x$, $h_2 = g^y$. We define $DH_h(h_1, h_2) = DH_g(g^x, g^y) = g^{xy}$ The discrete algorithm problem is to compute $log_g h$ The CDH problem is to compute $DH_g(h_1, h_2)$ The DDH problem is distinguish $DH_g(h_1, h_2)$ frome a uniform element of G. DDH > CDH > DLog

9

9.1

EI Gamal encryption scheme is not secure against the chosen ciphertext atatch. CCA-secure schemes are not malleable.

9.2

The El Gamal signature scheme using hash-then-sign paradigm is secure against the chosen plaintext attack.

9.3

Yes. we forge a signature for any given message m by asking the signing oracle.