Alex Berke May 1, 2023

6.5610 Problem Set 3

Collaborators: *Justin Yu, Jay Hilton, Fareed Sheriff*

Problem 1

(a) True. Collision resistance implies target collision resistance. Suppose H is not target collision resistant. Given x_1, and then given a random hk, an adversary can find x_2 s.t. $H(hk, x_1) = H(hk, x_2)$ with more than negligible probability. Then the adversary can also find two distinct x, namely x_1, x_2 s.t. $H(hk, x_1) = H(hk, x_2)$ with more than negligible probability, and hence isn't collision resistant.

(b) I got this wrong. Solution: True.

(c) False. Suppose we have random oracle hash function H which is one-way. No we construct H' s.t. given $x \in \{0,1\}^n, H'(x) = H(x)[0 : n-1]||0$. H' is one-way because if the probability of finding x given $H(x)$ is negligible then also the probability of finding x given $H'(x)$ is negligible. But H' is not collision resistant because we can choose x, x' s.t. they are the same for all but the last bit and then $H'(x) = H'(x')$.

(d) Yes H' is collision resistant. Call $H(hk, x) = y$ so $H'(hk, x) = H(hk, y) = H(hk, H(hk, x))$. Suppose $H'(hk, x) = H'(hk, x')$, i.e., $H(hk, y) = H(hk, y')$. Since we know H is collision resistant, we can assume $y = y'$. Since $H(hk, x) = y$ and $H(hk, x') = y'$, again since H is collision resistant we can assume $x = x'$ (with high probability). Thus H' is collision resistant.

(e) Yes H' is collision resistant. This is because H is collision resistant and H' is the same function as H, acting over the same domain space, except appends another constant bit. This extra bit does not change the collision resistance, it is simply inefficient by taking up more space without adding complexity.

(f) H' is not necessarily collision resistant. Suppose $H(hk, x) = 0^{k/2}||H"(hk, x)[k/2 : k]$ where $H" : \{0,1\}^* \rightarrow \{0,1\}^k$ is a random oracle hash function that has a collision with probability $\leq \frac{1}{O(2^{k/2})}$. First we note H can be collision resistant even with the leading $k/2$ 0's: there are collisions with probability $\leq \frac{1}{O(2^{k/4})} \leq \frac{1}{poly(k)}$. Given this construction of H, H' outputs 0's and is clearly not collision resistant.

Problem 2

(a) One-way function. Given any message $m \in \{0,1,\}^n$, there are the n corresponding $y_{m_i,i}$ in the pk that are used to verify the signature on this message. The adversary must find corresponding x_i s.t. $H(x_i) = y_{m_i,i}$ for each such $y_{m_i,i}$. If H is a one-way function the adversary can do this with less than negligible probability. Target collision resistance is not needed because the adversary does not already have the input x.

(b) No. Given $m' \neq m$, there is some bit i s.t. m', m differ in bit i. i.e. $m'_i \neq m_i$. Then given the construction of the signature, $\sigma'_i \neq \sigma_i$. Without loss of generality, suppose $m'_i = 0, m_i = 1$. To produce a valid σ', the adversary must find σ'_i s.t. $H(\sigma'_i) = y_{0,i}$. Since H is a one-way function, the adversary cannot do this with more than negligible probability.

(c) The adversary can choose $m_1 = 0^n, m_2 = 1^n$. From σ_1 the adversary learns all $x_{0,i}$ in sk and from σ_2 the adversary learns all $x_{1,i}$ in sk. The adversary then has all the information needed to sign any message.

(d) For message m of any length, first produce $h = H'(m) \in \{0,1,\}^n$. Then sign h just like m was signed above. i.e. $\sigma = (x_{h_1,1}, x_{h_2,2}, ..., x_{h_n,n})$ where h_i denotes the i-ith bit of h.

(e) Note my solution was graded as incorrect. The solution set used "signature chaining".

My solution: sk is chosen the same as before. But this time pk is computing by hashing each element in sk twice. i.e. $y_{0,i} = H(H(x_{0,i}))$ and $y_{1,i} = H(H(x_{1,i}))$ for $1 \leq i \leq n$. To sign the first message, $m1$, the scheme from (d) is used to produce $h1 = H'(m1)$. Then $\sigma 1 = (H(x_{h1_1,1}), H(x_{h1_2,2}), ..., H(x_{h1_n,n}))$. Verify outputs accept if $H(\sigma 1_i) = y_{h1_i,i}$ for all $1 \leq i \leq n$. Now that the values $H(x_{h1_i,i})$ have been revealed to sign a message, we do not use them again. To sign the second message, $m2$, again compute $h2 = H'(m2)$. Then $\sigma 2$ is computed exactly as in (d), s.t. $\sigma 2_i = x_{h2_i,i}$. Verify outputs accept if $H(H(\sigma 2_i)) = y_{h2_i,i}$ for all $1 \leq i \leq n$.

Problem 3

Since G is of prime order we assume g and h are both generators.

(a) Statistically hiding. For $x_1 \neq x_2$, then $g^{x_1} \neq g^{x_2}$. Since r_1, r_2 are sampled independently from a random distribution, the resulting h^{r_1}, h^{r_2} are also randomly distributed in G s.t. $\{h^{r_1}\} \equiv \{h^{r_2}\}$. The resulting commitments $g^{x_1}h^{r_1}, g^{x_2}h^{r_2}$ are then just as randomly distributed in G.

(b) Computationally binding, assuming the DLOG problem is hard. Call commitment $c = g^{x_1}h^{r_1}$ and $a = g^{x_1}, b = h^{r_1}$ s.t. $c = ab$. Since G is of prime order, g, h are generators, there exists some x_2 s.t. $g^{x_2} = b = h^{r_1}$ and some r_2 s.t. $h^{r_2} = a = g^{x_1}$ s.t. $g^{x_2}h^{r_2} = c$. An all powerful adversary could find such x_2, r_2 so commitments are not statistically binding. However, since G is large, given the discrete log problem is hard, we assume a computationally bounded adversary cannot find such x_2, r_2.

Problem 4

My kerberos: aberke

Secret: U$3 3nCrypt10n

Code: `https://github.com/aberke/applied-crypto-and-security-6.5610/blob/master/pset3/client.py`