Alan Liu
CPSC 467 Cryptography
November 2, 2017
Professor Michael J. Fischer

Problem Set 7

Problem 1: ElGamal Authentication

a. Describe why Bob accepts every message that Happy sends in this way (assuming no errors in transmission).

Operating under the assumption that the r was consistent and there are no external actors, H is sending $s = S(r) \oplus h(m \oplus r)$. I will validate that with this s, Bob will always validate it to be true by showing his validation algorithm returns true.

$$\begin{aligned} V(r, s \oplus h(m \oplus r)) &= V(r, (S(r) \oplus h(m \oplus r)) \oplus h(m \oplus r)) && \text{(replace s)} \\ &= V(r, S(r)) && \text{(invertability of xor)} \\ &= True && \text{(Definition of verification and signing functions)} \end{aligned}$$

Thus, Bob's validation algorithm will always accept Happy's message.
Q.E.D.

b. Mallory wants to replace m with a message m' of his choosing and get Bob to accept it as valid. Describe in detail how he can do this. Assume that Mallory is carrying out a man-inthe-middle attack, but she does not know Happys signing key and cannot forge signatures S(x) for messages x of Mallorys choosing.

I assumed that the hash function was public.

When r is sent, M can note down the value of r. Then when (m,s) is sent, M can replace it with $(m', (s \oplus h(m \oplus r)) \oplus h(m' \oplus r)$. B will then accept it as valid since it uses the same r as before, and it will evaluate to the same signature as the one H computed after applying the xor with the hash function of the received message.
Q.E.D.

c.Suggest a way to fix this protocol to thwart Mallorys attack. Your suggestion should not use any more rounds of communication nor assume any other encryption system or secret keys. Explain.

Let $s = S(h(m \oplus r))$. B would check $V(h(m \oplus r), s)$. Then the signature hides both the message and r, so they are bound together.
Q.E.D.

Problem 2: Hash from Cryptosystem

a. Given any k 1 and 128-bit string s_k, show how to find a message $M = m_1 m_2 ... m_k$ such that $H(M) = s_k$.

Since it suffices to find only one message M that hashes to this function, let us assume then that for all $i > 1$, $m_i = 0$. We can then calculate $D_0^{k-1}(s_k)$ (using the decryption function k-1 times) to get m_1. After concatenating them together, we have found an M that satisfies the condition that $H(m) = s_k$.

Q.E.D.

b. Show how to find a colliding pair (M, M') for H().

Let M_{p1}, M_{p2} be arbitrary messages of length p. Let s_k be an arbitrary string of length 128 bits and $H(M_{p1}) = s_{p1}, H(M_{p2}) = s_{p2}$. Since all strings s have the same length of 128 bits, we can xor them with a unique string to generate s_k using xor with s_k. Let $m_{k1} = s_{p1} \oplus s_k, m_{k2} = s_{p2} \oplus s_k$. We can then append m_{k1} to M_{p1} to make M, and m_{k2} to M_{p2} to make M'. We can then see that because $H(M) = H(M_{p1}) \oplus m_{k1} = s_k = H(M_{p2}) \oplus m_{k2} = H(M')$. Thus M, M' collide for H.

Q.E.D.