Crypto HW2b

beckej3

October 2018

Problem 1: Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and a primitive root $\alpha = 7$.

1a: If User A has private key $X_A = 5$, what is A's public key Y_A?
$Y_A = \alpha^{X_A} \text{mod } q = 7^5 \text{mod } 71 = 51$

1b: If User B has private key $X_B = 12$, what is B's public key Y_B?
$Y_B = \alpha^{X_B} \text{mod } q = 7^{12} \text{mod } 71 = 4$

1c: What is the shared secret key?
$K = Y_B^{X_A} \text{mod } q = Y_A^{X_B} \text{mod } q = 51^{12} \text{mod } 71 = 4^5 \text{mod } 71 = 30$

1d: In the Diffie-Hellman protocol, each participant selects a secret number x and sends the other participant ($\alpha^x \text{mod } q$) for some public number α. What would happen if the participants send each other ($x^\alpha \text{mod } q$) instead?

There are two problems with this. Firstly, If the recipient of the public key uses their own private key y to compute the shared key, The sender and receiver would most likely end up with different shared key values because $x^{\alpha^y} \bmod q$ does not necessarily equal $y^{\alpha^x} \bmod q$.

Secondly, if for some reason the participants sent each other $K = (x^\alpha \text{mod } q)$ as the public key, an adversary would realize that

$$\begin{aligned}
K &= x^\alpha \text{mod } q \\
x^\alpha &= K + nq \text{ where n is an integer} \\
x^\alpha - K &= nq \\
q &| (x^\alpha - K)
\end{aligned}$$

Finding a value for x which satisfies this condition is much easier than solving the discrete log problem, so the adversary could easily get the secret key x. Therefore, this protocol would be insecure in addition to being broken.

Problem 2: A network resource X is prepared to sign a message by appending the appropriate 64-bit hash code and and encrypting that hash code with X's private key as described in class (also in the textbook page 330).

2a: Describe the birthday attack where an attacker receives a valid signature for his fraudulent message

To obtain a valid signature for a fraudulent message, the attacker must first generate $2^{m/2}$ variations of a valid message, then generate $2^{m/2}$ fraudulent messages. Then, the attacker compares the two sets of messages to find a pair which have the same hash. By the birthday paradox, there exists such a pair of messages with probability greater than 0.5. The attacker then gets the valid message signed, and then substitutes the fraudulent message with the same valid signature.

2b: How much memory space does an attacker need for an M-bit message?

The attacker must store each message and its hash, which requires (M + 64) bits of storage for one message. There are 2^{32} valid message and 2^{32} fraudulent messages generated, so the attacker needs at least $2(2^{32}(M+64)) = 2^{33}(M+64)$ bits.

2c: Assuming that the attackers computer can process 2^{20} hashes per second, how long does it take on average to find a pair of messages with the same hash?

In the average case, the attacker must compute 2^{33} hashes. It therefore takes $\frac{2^{33}}{2^{20}} = 2^{13}$ seconds to find a match, which is 136 minutes and 32 seconds.

2d: Answer (b) and (c) when a 128-bit hash is used instead

The attacker must store $2^{65}(M+128)$ bits of data, and computing the hashes takes 2^{45} seconds, or roughly 1115689 years. This takes much more storage and much more time, so clearly the 128-bit hash is more secure.

Problem 4: Use Trapdoor Oneway Function with the following secrets as described in lecture notes to encrypt plaintext P = 0101 0111. Decrypt the resulting ciphertext to obtain the plaintext P back

First, compute the public key:

$$\begin{aligned}
t_1 &= 1019 \cdot 5 \bmod 1999 = 1097 \\
t_2 &= 1019 \cdot 9 \bmod 1999 = 1175 \\
t_3 &= 1019 \cdot 21 \bmod 1999 = 1409 \\
t_4 &= 1019 \cdot 45 \bmod 1999 = 1877 \\
t_5 &= 1019 \cdot 103 \bmod 1999 = 1009 \\
t_6 &= 1019 \cdot 215 \bmod 1999 = 1194 \\
t_7 &= 1019 \cdot 450 \bmod 1999 = 779 \\
t_8 &= 1019 \cdot 946 \bmod 1999 = 456
\end{aligned}$$

So the public key is T = {1097, 1175, 1409, 1877, 1009, 1194, 779, 456}
Now, compute Y = (1175+1877+1194+779+456) = 5481
Y = 5481 is the ciphertext. To decrypt it, compute

$$Z = Y \cdot a^{-1} \bmod p = -410(5481) \bmod 1999 = 1665$$

Now, simply use the original superincreasing list to solve the instance of the subset problem I(S, Z). Notice that

$$1665 = (0)5 + (1)9 + (0)21 + (1)45 + (0)103 + (1)215 + (1)450 + (1)946$$

The coefficients in this expression are 01010111, which is the original plaintext.