# Cryptography - Homework 2c

Samuel Breese

October 15, 2018

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

## 1.1 If user A has a private key $X_A = 5$, what is $A$'s public key $Y_A$?

$Y_A = \alpha^{X_A} \mod q = 7^5 \mod 71 = 51$

## 1.2 If user B has a private key $X_B = 12$, what is $B$'s public key $Y_B$?

$Y_B = \alpha^{X_B} \mod q = 7^{12} \mod 71 = 4$

## 1.3 What is the shared secret key?

$K = Y_A^{X_B} \mod 71 = X_A^{Y_B} \mod 71 = 51^{12} \mod 71 = 30$

## 1.4 In the Diffie-Hellman protocol, each participant selects a secret number x and sends the other participant $(\alpha^x \mod q)$ for some public number $\alpha$.

**What would happen if the participants sent each other $(x^\alpha \mod q)$ instead?**

This modification means that an attacker trying to obtain the private key from the public key would only need to compute a modular root, not a discrete logarithm. Efficient algorithms exist to compute such roots, and thus all security is lost - our function is no longer one-way.

# 2 A network resource $X$ is prepared to sign a message by appending the appropriate 64-bit hash code and encrypting that hash code with X's private key as described in class.

## 2.1 Describe the *Birthday Attack* where an attcker receives a valid signature for his fraudulent message?

An attacker would generate $2^{32}$ valid-looking messages and $2^{32}$ fraudulent messages, and would compute the hash of each of these. By the birthday paradox, the probability that a collision occurs between at least one valid/fraudulent pair of messages is greater than 0.5. The attacker would then ask the network resource $X$ to sign the valid-looking message, and attach the encrypted signature to the corresponding fraudulent message. The fraudulent message now has a valid signature (since its hash value is the same as the valid message).

## 2.2 How much memory space does attacker need for an M-bit message?

During the attack, the attacker needs to store $2 \times 2^{32}$ messages, each of length $M$ bits. The attacker must also store the 64-bit hash code for each of those messages. As such, the attacker requires $(M + 64) \times 2^{33}$ bits of memory to perform the birthday attack.

## 2.3 Assuming that attacker's computer can process $2^2 0$ hash/second, how long does it take (on average) to find a pair of messages that have the same hash?

Again, the attacker must hash $2^{33}$ messages, so the attack will take

$$\frac{2^{33} \text{ msg}}{2^{20} \text{ msg/s}} = 2^{13} \text{ s}$$

This is only about two hours and 15 minutes.

## 2.4 Answer the previous two questions when 128-bit hash is used instead.

The attacker will now need $(M + 128) \times 2^{65}$ bits of memory, and $2^{45}$ seconds of computation time (this is more than one million years).

## 3 Use *Trapdoor Oneway Function* with following secrets as described in lecture notes to encrypt plaintext $P = 01010111$.

**Decrypt the resulting ciphertext to obtain the plaintext $P$ back. Show each step to get full credit.**

$$S = \{5, 9, 21, 45, 103, 215, 450, 956\}$$
$$a = 1019, \ p = 1999$$

Assuming this describes the Merkle knapsack system, we must first compute the "hard" knapsack from

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$$

To do this, we compute $h_i = a(s_i) \mod p$ for all $1 \leq i \leq 8$. This yields

$$H = \{1097, 1175, 1409, 1877, 1009, 1194, 779, 651\}$$

From here, we can encrypt a message by summing the elements $h_i$ of $H$ where the corresponding bit $p_i$ in $P$ is one:

$$C = 1175 + 1877 + 1194 + 779 + 651 = 5676$$

To decrypt, we must first find the modular inverse of $a \mod p$, which is easy using Fermat's Little Theorem since 1999 is prime:

$$a^{-1} = 1019^{1997} \mod 1999 = 1589$$

Since multiplication distributes over addition, we can get the sum of elements in $S$ by computing:

$$a^{-1}C \mod p = (1589)(5676) \mod 1999 = 1675$$

We can then easily solve the subset sum problem on $S$ and $a^{-1}C$ with a greedly algorithm:

1. Given 1675, notice that $956 \leq 1675$. The eighth bit of the plaintext is therefore 1, and $1675 - 956 = 719$.

2. Given 719, notice that $450 \leq 719$. The seventh bit of the plaintext is therefore 1, and \$719 - 450 $=$ 269%.

3. Given 269, notice that $215 \leq 269$. The sixth bit of the plaintext is therefore 1, and $269 - 215 = 54%.

4. Given 54, notice that $103 \not\leq 54$. The fifth bit of the plaintext is therefore 0.

5. Given 54, notice that $45 \leq 54$. The fourth bit of the plaintext is therefore 1, and $54 - 45 = 9$.

6. Given 9, notice that $21 \not\leq 9$. The third bit of the plaintext is therefore 0.

7. Given 9, notice that $9 \leq 9$. The second bit of the plaintext is therefore 1, and $9 - 9 = 0$.

8. The first bit of the plaintext must be 0, since the subset sum has been found.

Thus, the decrypted ciphertext is 01010111 which is the same as the plaintext.