Joud Mawad (Matrikelnummer: 5377552)

Luis Jair Gutierrez Pacheco (Matrikelnummer: 5453416)

# Diffie-Hellman

## 1. Alice, Bob and Carol want to use a shared secret key for symmetric encryption.

### (a)

- Each person will choose a private exponent, so Alice chooses a, Bob chooses b and Carol chooses c and all will use the modulo n which is shared and not private.
  We have also a generator g which is also not private and each person will use to compute the public value.

  Alice will compute $A = g^a\, mod(n)$ and will send A to Bob and Carol

  Bob will compute $B = g^b\, mod(n)$ and send it to Alice and Carol

  Carol will compute $C = g^c\, mod(n)$ and sends it to Alice and Bob

  Now Alice would use her secret value a on the second persons value let's say B value from Bob and will send the result to Carol, in other words:

  $X = B^a = g^{ba}\, mod(n)$ then send X to Carol.

  The same goes for Bob: $Y = C^b = g^{cb}\, mod(n)$ and will send it to Alice

  and for Carol $Z = A^c = g^{ac} mod(n)$ and will send it to Bob

  now Alice have $Y = g^{cb}$ and can Compute $K = Y^a = g^{cba}\, mod(n)$
  Bob would compute $K = Z^b = g^{acb}\, mod(n)$
  Carol would compute $K = X^c = g^{bac}\, mod(n)$

  Now all of them have the same Key.

### (b)

- Public Stuff:
  In Diffie-Hellman all people use the same $n$ and the same generator $g$, and each one just picks a private number $x_i$. In RSA every person has their own public key $(e_i, N_i)$, so we have $n$ different keys.

- Messages:
  For Diffie-Hellman every person sends $g^{x_i}\, mod(n)$ and later sends one more combined value, so it is $O(n)$ messages and it needs interaction.
  For RSA one person can just encrypt the shared key $K$ for every other user and send $n$ messages, also $O(n)$, but no interaction.

- Forward Secrecy:
  Diffie-Hellman gives forward secrecy if everyone uses fresh private numbers, because these numbers get deleted later. RSA does not have forward secrecy, because if someone gets one private key $d_i$ they can decrypt all old messages.

## 2. Alice and Bob use the Diffie-Hellman key exchange to negotiate a common shared key.

### (b)

- To derive the secret shared key, we need to solve the Discrete Logarithm Problem. Given the public values $g$, $n$, and Alice's public key $A = g^a modn$, we need to find the private exponent $a$.
- Normally this is a real challenge for large numbers. In this example, $n$ is really small, which allows us to perform a Brute Force attack, simply trying every possible integer for the exponent until the correct public value is found.

## 3. Messages corresponding to the numbers 0, 1 and N-1 have a special property...

Messages corresponding to 0, 1, and $N-1$ are known as **fixed points** in RSA. This means that when encrypted, the ciphertext is identical to the plaintext ($c = m$).

The RSA encryption function is defined as $c \equiv m^e \pmod{N}$.

- For $m = 0$:
$$0^e \equiv 0 \pmod{N}$$
(Since 0 raised to any positive integer power remains 0).
- For $m = 1$:
$$1^e \equiv 1 \pmod{N}$$
(Since 1 raised to any power is always 1).
- For $m = N - 1$: In modular arithmetic, $N - 1 \equiv -1 \pmod{N}$.
$$c \equiv (N-1)^e \equiv (-1)^e \pmod{N}$$
Since the public exponent $e$ in RSA is typically an odd integer (to be coprime with $\varphi(N)$):
$$(-1)^e = -1 \equiv N - 1 \pmod{N}$$
Therefore, the ciphertext remains $N - 1$.

## 4. Master: Alice uses the RSA algorithm for encrypting a message m...

### (b)

- We need to solve the Integer Factorization Problem.
- To decrypt the message, we need the private key $d$, which is the modular multiplicative inverse of $e$ modulo $\varphi(n)$. To calculate $\varphi(n) = (p-1)(q-1)$, we must know the prime factors $p$ and $q$ of the modulus $n$.
- Since the modulus provided is small, we can factorize it easily to find $p$ and $q$.