# Multiple parties Diffie-Hellman

Tufa Alexandru Daniel 1628927

10 November 2017

## 1 Introduction

Diffie-Hellman key exchange, we'll often abreviate it as DH, is a way of exchanging public information to arrive at che creation of a common private key that will be used to encrypt communications using a symmetric key cipher. The public parameters used between two parties, for simplicity called Alice and Bob, are a prime p, that should be of at least 600 digits and a safe prime ( p = 2q + 1 ), and an element g ( possibly a generator of the multiplicative group ${Z_\mathrm{p}}^*$ ) that can be small because the underlying function that we'll be using, the discrete logarithm function, has the property of random self-reducibility[1]. Alice and Bob choose their private key randomnly belonging to the interval [1,..,p-1] and Alice, after calculating A = $g^a$ mod p sends to Bob g, p and A. Bob, using his private key, calculates B = $g^b$ mod p, the private key S = $A^b$ mod p and sends to Alice B so that she can also create the private key S = $B^a$ mod p. The two private keys are equal because S = $A^b$ mod p = $g^{ba}$ mod p = $B^a$ mod p. We can be sure that the key is safe because in order to compute the private key of Alice or Bob the attacker should be able to compute a or b by knowing g, p and A or B and this problem is called the discrete logarithm problem that is considered to be a `hard`[2] problem. So this methodology is considered safe against a passive attack, while it's unsecure again a Man-In-The-Middle-Attack because the attacker could intercept all the messages between Alice and Bob and establish two distinct key exchanges so that he can encrypt and decrypt messages passed between them. However if the attacker is ever absent this would reveal to the two parties that their communication has been compromised. To prevent this an authentication method should be used.

[1] If an algorithm is good for the average case it implies it's also good for the worst case

[2] If P $\neq$ NP

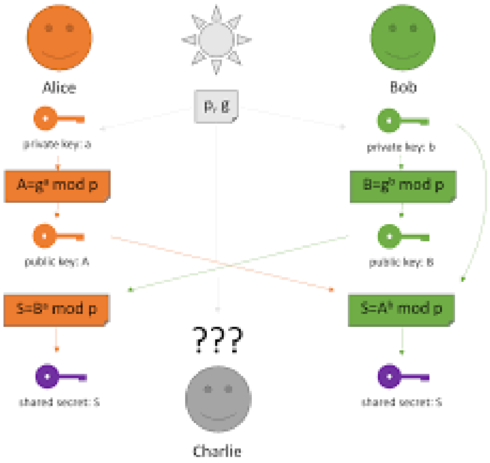

Figure 1: Diffie-Hellman key exchange

## 2 Description

We need to analyze the configuration of this protocol in the case the parties involved in the DH key exchange are more than two. Initially we'll consider the case where there are three members involved and later generalize it to n parties.

## 3 Analysis

### 3.1 Three members

For three members we'll use a simple ring configuration where every party has a left and a right member. This configuration can be extended even to the general case, but we'll see a better implementation. In this case the parties agree on the public parameters p and g and choose their private keys a, b and c. Alice starts the protocol by sending to Bob A=$g^a$ mod p. Bob computes AB[3] = $g^{ab}$ mod p and B = $g^b$ mod p and sends them to Carol. Carol computes $S = (AB)^c$ mod p that keeps for herself as secret key and computes BC = $g^{bc}$ mod p and C = $g^c$ mod p and sends them to Alice. Alice can create the private key $S = BC^a$ mod p and computes CA=$C^a$ mod p that is sent to Bob who calculates the secret key $S = CA^b$ mod p. As we can see for three members four messages have been used.

[3] AB is not A*B but it's a notation for representing $g^{ab}$ mod p

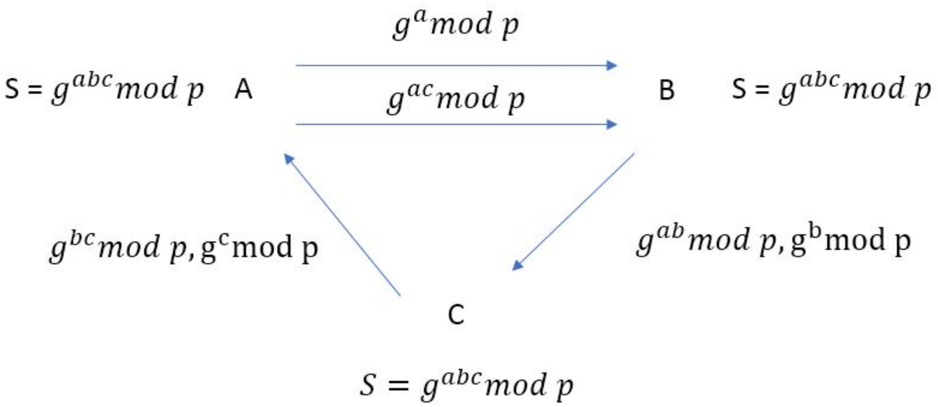

Figure 2: Three parties key exchange

### 3.2 General considerations

We can extend this reasoning even if we consider n members and we'll use a total of n + n - 2 messages because we need to pass all the exponentiations done using the private key of each member and also the composition of the past exponentiations with each members own private key. Because we follow the ring model the member before the one that has sent the first message will have computed the secret private key and we have to continue transmitting the exponentiation of this member to the one before it and to do this we need n - 2 messages.
In every case this methodology will always be vulnerable to MITM[4] attacks without an authentication protocol. For n parties the attacker would need a total of

$$\frac{n(n-1)}{2}$$

keys to intercept all communications between them. If only one member is compromised during the creation of the private key then the attacker can create itself a private key with the other members and then instaurate a two parties DH key exchange between himself and the compromised member. He will receive all messages incoming from the others,modify and send them impersonating every time the sender of the message. By doing this the attacker has the power of observing all the network and to prevent this we

[4] Man In The Middle

must absolutely use an authentication protocol.
In case of a crash failure and a ring configuration, during the creation of the secret key all processes stop because every member awaits information from its left neighbour ( if we have a clockwise communication ). However this can be resolved by using a Perfect-Failure Detector[5] to ensure that no process remains without a neighbour during the key exchange. If the communication is not considered a perfect link where we are certain that every sent message will be received we can implement an additional software layer and send the message many times until we receive an acknowledgment from the receiver. If this is not possible then our considerations are analogous to the ones made in the case of a crash failure because a missing message would stop all the key exchange protocol.

### 3.3 Alternative implementation

We have seen that in a ring configuration every participant is required to perform n modular exponentiations. By using a divide and conquer approach we can reduce this number to $log_2(\mathrm{n}) + 1$ exponentiations. We can imagine dividing the network in half every time and communicating each time the resulting value of one network to the other one and vice versa. Suppose we have N members $\mathrm{n}_1$,...., $\mathrm{n}_\mathrm{N}$ with secret keys $\mathrm{m}_1$,..., $\mathrm{m}_\mathrm{N}$. The members $\mathrm{n}_1$,...,$\mathrm{n}_{\mathrm{N}/2}$ each perform one exponentiation sending to the members of the other half $g^{m_1 m_2 .. m_{\mathrm{N}/2}}$ mod p while they receive the value $g^{m_{\mathrm{N}/2+1} m_{\mathrm{N}/2+2} .. m_\mathrm{N}}$ mod p. This procedure is performed recursively until each member does the final exponentiation and instead of N exponentiations this number is reduced to $log_2(\mathrm{n}) + 1$.

[5] An oracle that notifies if a neighbour has failed and provides a new one