Overview of Diffie-Hellman key exchange

Alexander Shevtsov
Magic Department, Hogwarts University

June 17, 2016

Contents

1 Introduction

Most of the cryptographic algorithms use some sort of secret information that is used for encryption and decryption. That information must be known only to parties involved in communication. Even if the eavesdropper manages to understand which encryption algorithm is used, this secret information (usually called key or password) prevents him from decrypting the message.

The Diffie[1]-Hellman[2] key agreement protocol (published in 1976) was the first practical method for establishing a shared secret over an insecure communication channel.

Prior to that time, all useful modern encryption algorithms had been symmetric key algorithms, in which the same cryptographic key is used with the underlying algorithm by both the sender and the recipient, who must both keep it secret. All of the electromechanical machines used in WWII were of this logical class, as were the Caesar and Atbash ciphers and essentially all cipher systems throughout history. The 'key' for a code is, of course, the codebook, which must likewise be distributed and kept secret, and so shares most of the same problems in practice.

Of necessity, the key in every such system had to be exchanged between the communicating parties in some secure way prior to any use of the system (the term usually used is 'via a secure channel') such as a trustworthy courier with a briefcase handcuffed to a wrist, or face-to-face contact, or a loyal carrier pigeon. This requirement is never trivial and very rapidly becomes unmanageable as the number of participants increases, or when secure channels aren't available for key exchange, or when, as is sensible cryptographic practice, keys are frequently changed. In particular, if messages are meant to be secure from other users, a separate key is required for each possible pair of users. A system of this kind is known as a secret key, or symmetric key cryptosystem. D-H key exchange (and succeeding improvements and variants) made operation of these systems much easier, and more secure, than had ever been possible before in all of history.

[1] Bailey Whitfield Diffie (born June 5, 1944), an American cryptographer

[2] Martin Edward Hellman (born October 2, 1945), an American cryptologist

2 Mathematical preliminaries

Perhaps the only mathematical object you need to know is the concept of group.

Definition. A group G is a non-empty set with a defined binary operation $(\times)$ called composition that obeys following laws:

1. $(a \times b) \times c = a \times (b \times c) \quad \forall a, b, c \in G$ (associativity of composition).
2. There exists an identity element of G denoted as e, such that composition of any element a with identity element produces a:
$$\forall a \in G \quad a \times e = a$$
3. For any element a there exists inverse element: $\forall a \in G\ \exists (a^{-1}) : a \times (a^{-1}) = e$.

The definition of binary operation

$$\times : G \times G \to G$$

means that by taking two elements of group we can produce another element of group. So the group is closed under composition operation: the result of composition is some element from the group.

Notation a^n for an element a of some group G and some integer n means:

$$a^n = \underbrace{a \times a \times \ldots \times a}_{\text{n times}}.$$

For negative n it's defined as follows:

$$a^{(-n)} = (a^{-1})^n.$$

Element in zeroth power is defined as usual:

$$a^0 = e.$$

Definition. A group G with binary operation $\times$ is called *commutative* or *abelian* if for any two elements a, b of the group holds:

$$a \times b = b \times a$$

It's clear, that instead of using symbol × for defined binary operation we can use any other symbol, for example $+$, or $\cdot$. Traditionally symbol $+$ is used for abelian groups and such groups are called *additive* groups, in that case the identity element is denoted as 0 and the inverse element of a is denoted as $-a$.

Symbol $\cdot$ is used for non-abelian groups, these groups are called *multiplicative*, the identity element is sometimes denoted as 1. Often the composition operation in multiplicative groups is omitted, so $a \cdot b = c$ becomes just $ab = c$.

For abelian groups power notation transforms as follows: $a^n = a \times a \times a \ldots \times a$ becomes $a + a + \cdots + a = na$ (yes, that may be confusing).

Examples of groups: integer numbers $\mathbb{Z}$ with the respect to addition, same for real numbers $\mathbb{R}$ and complex numbers $\mathbb{C}$.

When some set has several group structures simultaneously, considered group structure is understood from context or explicitly expressed, for instance $(\mathbb{R}, +)$ shows that we are going to view reals numbers as a group with respect to addition.

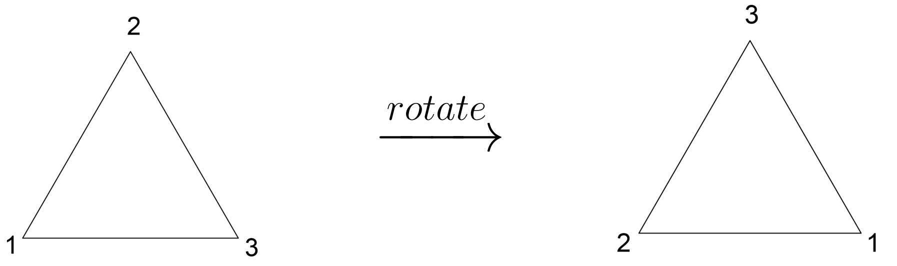
Exercise 1

Above examples are all abelian, give an example of non-abelian group.

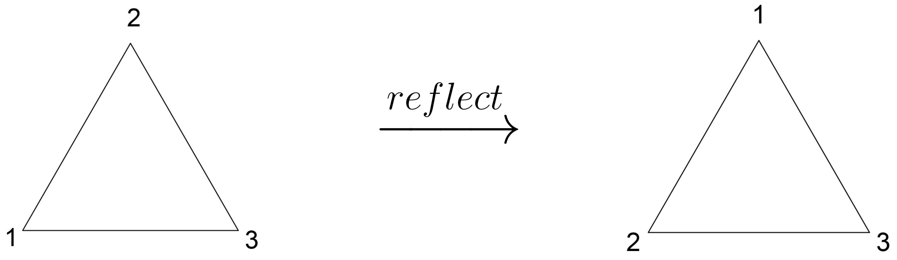
Answer of exercise 1

One of the most common examples of groups is a group of symmetries of some geometric obejct. For example consider a equilaterial triangle in a plane. All bijections of a plane that map vertices into vertices and edges into edges form a group *dihedral* group D_3. Indeed, the composition of a two such bijections is again a bijection that maps vertices to vertices and edges to edges, all other group axioms hold too. This group consists of 6 elements and is not abelian.

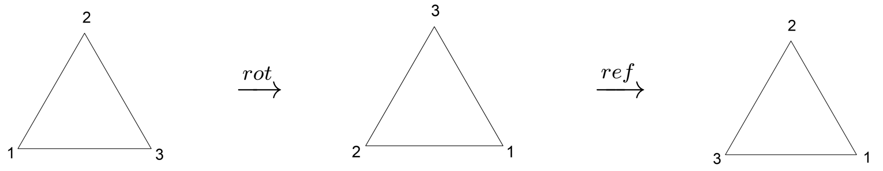
Consider a clockwise rotation.



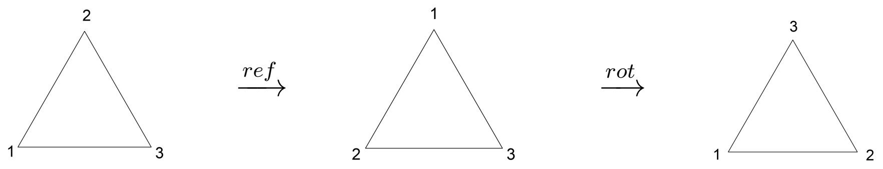
Consider reflection with respect of median from bottom right vertex.



Let’s compose these operations.



Composition taken in reverse order.



Composition of rotation with reflection does not equal to itself taken in the reverse order.

Exercise 2

Do integers form a group with the respect to multiplication operation?

Answer of exercise 2

No, because the only elements for which inverse elements exist are 1 and -1.

Further we will consider only groups with finite number of elements.

Some more definitions:

Definition. Generators $g_1, g_2, \ldots, g_n$ of a group G are elements, such that any other element x of a group can be written in form:

$$x = g_1^{n_1} g_2^{n_2} \ldots g_n^{n_k}.$$

So generators are sort of constructing blocks for a group. A group is called cyclic, if it can be constructed with a single generator g. That means that any element x can be obtained by composing g with itself:

$$\forall x \in G\, \exists n \in \mathbb{Z} : g^n = x.$$

Cyclic groups may have more than one generator, for example $(\mathbb{Z}, +)$ has two generators 1 and -1. Indeed, any integer can viewed as sum of ones:

$$a = \underbrace{1 + 1 + \cdots + 1}_{\text{a times}} = \underbrace{-((-1) + (-1) + \cdots + (-1)}_{\text{a times}}.$$

We will use group $(\mathbb{Z}_p, \cdot)$. It consists of nonzero elements of integers modulo some prime number p. Composition operation is defined naturally:

$$(a \mod p) \cdot (b \mod p) = (a \cdot b) \mod p.$$

This group is called multiplicative group of integers modulo p. Generators of this group are integers $\neq 1$.

Exercise 3

Compute $123456 \cdot 654321$ in $\mathbb{Z}_{1000003}$.

Answer of exercise 3

611039.

It's all mathematical information you need.

Additional information can be found in any book about abstract algebra, for example [4].

3 Diffie-Hellman key exchange [3]

3.1 Analogies

Let's start with some analogies. At first imagine Alice wants to send Bob a box with something valuable via post. But Alice considers post unreliable and potentially someone involved in delivery can steal the inside. The structure of the box allows Alice to lock it with a padlock.

What should she do?

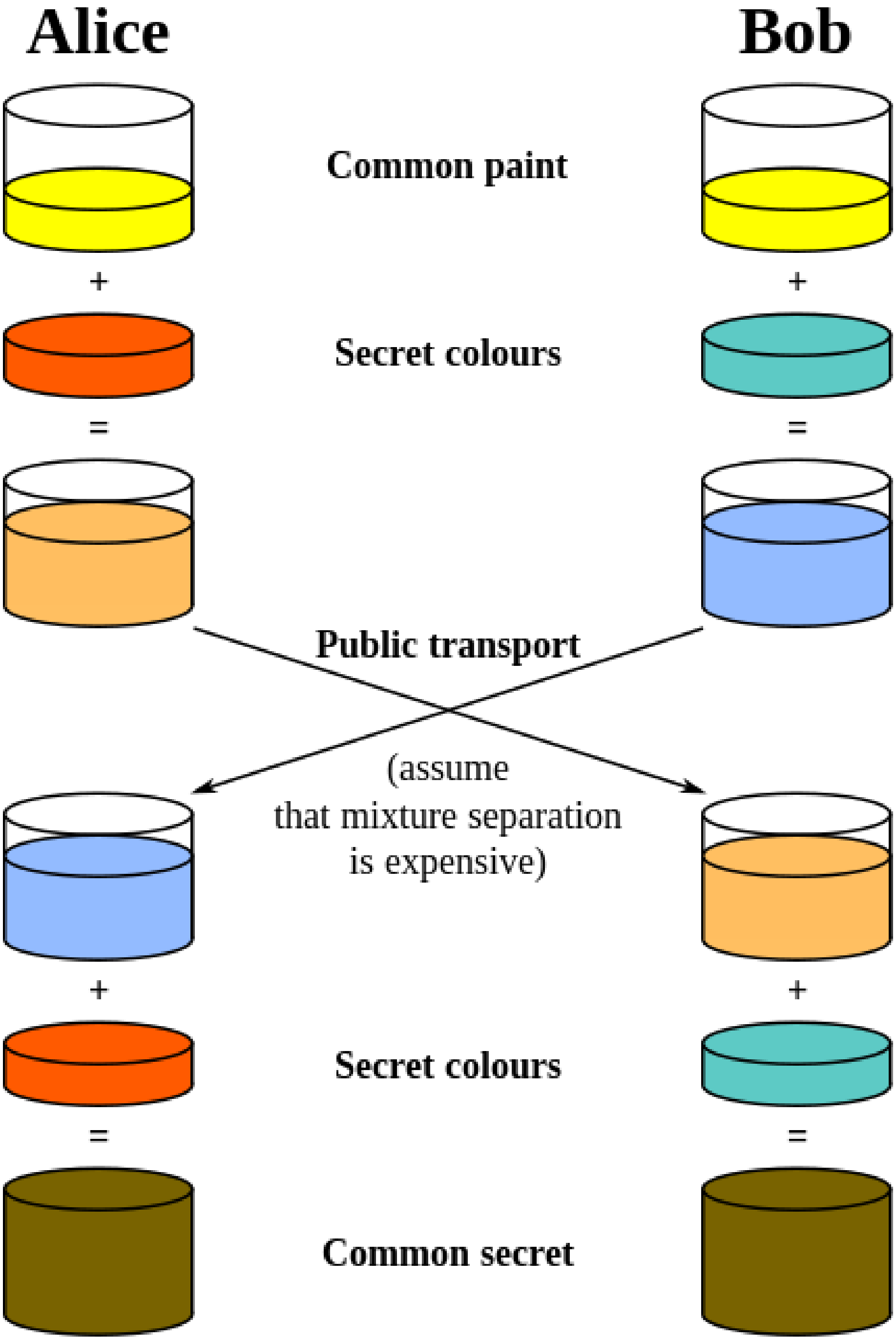
If she simply locks a box with her padlock, Bob will not be able to unlock it and to take the inside. However, there is a secure method of delivering the box, it only requires Bob to possess another padlock. It can be done as follows:

1. Alice locks a box using her padlock and sends it to Bob.
2. Bob receives a locked box and locks it with his padlock and sends back to Alice.
3. Alice receives a box locked by two padlocks, Alice unlocks her padlock and sends box back to Bob.
4. Bob receives a box with only his padlock, which he can unlock and open the box.

Using these steps you can perform a transaction via insecure channel. In that analogy post plays the role of an insecure channel, locking padlocks plays role of computations made in D-H key exchange, cracking padlock plays the role of a "difficult" task and the inside of the box plays the role of the shared secret (imagine that the inside is just a paper with a written password). However, it's not the closest analogy, as this analogy requires more transactions than D-H exchange.

Let's view another analogy. In that one the role of the shared secret plays the colour of the mixture. The process is illustrated in the image below:

[3] Authors suggest to call it Diffie-Hellman-Merkle key exchange in recognition of Ralph Merkle's contribution. Unfortunately, the title "Diffie-Hellman key exchange" is already too widely spread.



1. Alice and Bob choose a colour using an insecure channel (for example via telephone call) and both prepare a mixture of a chosen colour.
2. Alice and Bob (secretly) choose some arbitrary, probably random colour and prepare a mixture of this new colour and mix it with previously

obtained mixture.

3. Alice and Bob send via insecure channel to each other obtained mixtures (leaving some amount for themselves for further actions).

4. Each of the recipient mixes the received mixture with his own one, so both parties obtain the same colour.

In that analogy the "difficult" task is to obtain components of the mixture.

3.2 Implementation

Implementation depends on which group we're going to use. In the most simplest case the multiplicative group of integers modulo prime p is used, but D-H key exchange can use other groups too.

1. Alice and Bob via insecure channel choose a big prime number p, for example $p = 3183$.

2. Alice and Bob via insecure channel choose a generator g of group $\mathbb{Z}_p$, for example $g = 2$.

3. Alice chooses her secret number a, for example $a = 1000$. Alice computes $A = g^a$.

```
p=3183;
g=2;
a=1000;
A=PowerMod[g,a,p]
(*A=3100*)
```

4. Bob chooses his secret number b, for example $b = 10000$. Bob computes $B = g^b$.

```
b=10000;
B=PowerMod[g,b,p]
(*B=772*)
```

5. Alice and Bob send to each other A and B via insecure channel.

6. Alice computes her shared secret $Q = B^a$.

```
Q=PowerMod[B,a,p]
(*Q=961*)
```

7. Bob computes his shared secret $W = A^b$.

   ```
   W=PowerMod[A,b,p]
    (*W=961*)
   ```

At the final step both Alice and Bob computed the same number (in our case it's 961).

4 How does it work?

Let's review the algorithm.

1. Why do we choose prime p?

   We want to construct a finite (multiplicative) group of nonzero integers modulo p. They form a group if and only p is prime[4]. Bigger we take p more difficult is cracking of the algorithm, details will be discussed in security section.

2. Generators and exponentiation.

   Usually generators are not too large, however it can be any integer in $\mathbb{Z}_p$ except 1. Computing power of an integer can be done via fast exponentiation as described below.

   Assume we got two integers x, y. For example $x = 5673345$ and $y = 90987236$. Naïve method will take $y - 1 = 90987235$ operations of multiplication x by itself. Instead we can decompose $y = 2 \cdot 45493618$ and compute

   $$x^y = x^{90987236} = x^{2 \cdot 45493618} = (x^2)^{45493618}.$$

   Of course, we can continue process and decompose $y = 2 \cdot 2 \cdot 22746809$, then

   $$x^y = (x^{2^2})^{22746809}.$$

   At that step we got a power that is not divisible by 2, but we can extract one multiplication:

   $$x^y = (x^{2^2})^{22746808+1} = x^{2^2} \cdot (x^{2^2})^{22746808}$$

   and so on. This process requires roughly $\log_2 y$ multiplications, the value is:

   $$\log_2 y = \log_2 90987236 \approx 26.$$

   This method uses ninety millions multiplications less! For performing exponentiation modulo p we can optimize this algorithm even further: at each step of squaring we can reduce resulting integer modulo p.

[4] Actually we could consider non-prime p and take a multiplicative subgroup of invertible elements in $\mathbb{Z}_p$, in that case we should accurately take a generator of that group, it must be an integer comprime to p.

3. Shared secret.

   Both Alice and Bob obtain the same final result, because

   $$Q = B^a = (g^b)^a = g^{ba} = (g^a)^b = A^b = W.$$

4. Eavesdropper's view.

   Imagine there is an eavesdropper named Eva who can bug the insecure channel. She possesses values of p, g, A, B, but that's not enough to compute common secret $A^b = B^a$. That is, task of computing g^{ab} knowing only g, g^a, g^b is computationally difficult and it's the heart of Diffie-Hellman key exchange algorithm security.

5 Security

5.1 Diffie-Hellman problem

The security of Diffie-Hellman key exchange is based on complexity of Diffie-Hellman problem. It can be stated as follows:

Given elements g, g^a, g^b of some group, find the value of g^{ab}.

Today the most efficient method to solve it is to solve the discrete logarithm problem.

For additional information regarding Diffie-Hellman problem an problems equivalent to it, see [5].

5.2 Discrete logarithm problem

Discrete problem is highly related to Diffie-Hellman problem, but it is more general. It can be stated as follows:

Given elements g, a of some group find the integer n (if it exists), such that $g^n = a$. On the contrary to the case of real or complex numbers, where log can be computed using for instance Taylor series, in finite groups there is no general algorithm to solve that problem. If considered group G is not cyclic, then no solution may exist for some g and a, however in our case an integer n always exists (it's not unique), because we used cyclic group $\mathbb{Z}_p$ where every element is the generator g raised to some power.

The most trivial algorithm is to raise g to various powers until we get the needed value a, but it's too slow, because it involves numbers of operations roughly equal to the number of the group elements. There are some more sophisticated algorithms, but none of them is polynomial (in the number of digits in the size of groups).

Some of them:

- Baby-step giant step. [3]
- Pollard's rho. [1]
- Function field sieve. [2]

It's important to note, that there is no solution for Discrete logarithm problem (yet), however *it's not proven* that could be no computationally fast algorithm for solution of this problem. So, the security of Diffie-Hellman key exchange (and some others cryptographic algorithms) is based on *assumption* that such algorithm does not exist.

5.3 Man in the middle.

Pure Diffie-Hellman key exchange is vulnerable to attack known as "man in the middle". Consider the situation when the eavesdropper (Eve) can replace messages of the communication participants with her own messages:

1. Alice chooses secret a, computes $A = g^a$ and sends it to Bob.
2. Eve intercepts message of Alice and receives A. Then she chooses her secret e, and sends Bob $E = g^e$, imitating Alice.
3. Bob chooses secret b, computes $B = g^b$ and sends it to Alice.
4. Eve intercepts message of Bob and receives B.

Now Eve is able to compute g^{be} and g^{ae} and to imitate Bob and Alice. Neither Bob nor Alice will know they are communicating with Eve, while Eve can not only read encrypted messages of both Alice and Bob, but additionally she can write her own messages mimicking Bob or Alice.

That's why key exchange is used with additional security measures, for example:

- usage of certificates;
- time examination (if it normally takes say 20 seconds to make a key exchange, but the answer is received in 40 seconds that could mean there is a man in the middle);
- some method of online transmitting.

6 Additional information

6.1 Usage with more than two parties

Diffie-Hellman key exchange can be used for arbitrarily large number of participants, let's see a brief example for three of them, case for more number of participants is similar.

1. Alice, Bob and Carol publicly choose prime number p and generator g of group $\mathbb{Z}_p$.
2. Alice, Bob and Carol compute their own secrets: $A = g^a$, $B = g^b$, $C = g^c$.
3. Alice sens her secret to Bob and Carol, Bob computes $A^b = g^{ab}$, Carol computes $A^c = g^{ac}$. Bob sends g^{ab} to Carol and Carol sends g^{ac} to Bob.
4. Bob computes $(g^{ac})^b$ and it's his shared secret, same for Carol.
5. Carol sends Bob g^c.
6. Bob computes $(g^c)^b$ and sends it to Alice.
7. Alice computes $(g^{cb})^a$ and it's shared secret.

An eavesdropper has been able to see g^a, g^b, g^c, g^{ab}, g^{ac}, and g^{bc}, but cannot use any combination of these to efficiently reproduce g^{abc}.

6.2 In the wild

For secure usage prime numbers used in key exchange should be at least 2048 bit long (it's approximately 620 decimal digits).

Successful attacks solving discrete logarithm problem were performed for $p \approx 2^{9234}$ (it's about 2780 decimal digits), however it took equivalent of 400000 hours.

Computing hardware speed is always increasing, so if you want to use Diffie-Hellman key exchange that would be secure in next few years, it's recommended to use at least 4096-bit long prime number p. Records of solving discrete logarithm problem can be seen for example here.

References

[1] Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A. (2001). ”Chapter 3” (PDF). Handbook of Applied Cryptography.

[2] Leonard M. Adleman and Ming-Deh A. Huang. 1999. Function field sieve method for discrete logarithms over finite fields. Inf. Comput. 151, 1-2 (May 1999).

[3] A. Stein and E. Teske, Optimized baby step-giant step methods, Journal of the Ramanujan Mathematical Society 20 (2005), no. 1, 1–32.

[4] Serge Lang. Algebra. Springer-Verlag, New York, 2002.

[5] Feng Bao; Deng, Robert; Huafei Zhu (2003). ”Variations of Diffie–Hellman problem”. ICICS ’03 (Springer-Verlag) 2836: 301–312.