

Kryptographische Protokolle

The Decision Diffie-Hellman Problem

Matteo Harutunian
harutuni@in.tum.de
Technische Universität München

June 23, 2011

1 Introduction

The Diffie-Hellman key agreement protocol, is a procedure that allows establishing a shared secret over an insecure connection and was developed by Whitfield Diffie and Martin Hellman in 1976, . This procedure is, often slightly modified, still used today by several applications. Before going into more detail on these applications, let's have a look at the basic key-exchange method between two users Alice and Bob, where Alice is the one initiating the exchange:

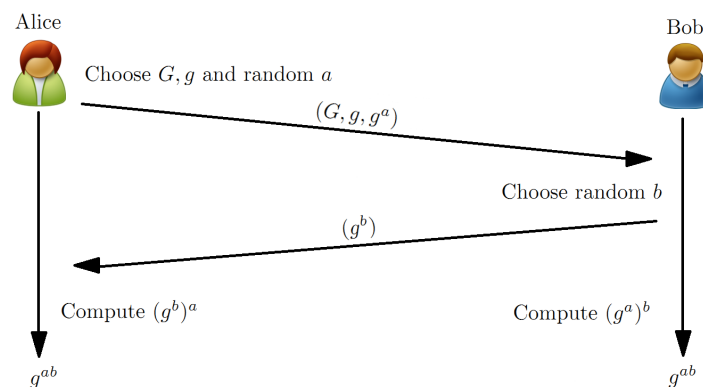




Figure 1: Diffie-Hellman key agreement protocol¹

First of all, Alice chooses the public parameters G and g and her secret parameter a . The limitations to her choices will be discussed in detail in section 3. She sends the triplet (G, g, g^a) to Bob. Upon receiving this triplet, Bob chooses his secret parameter b and sends g^b to Alice. Now Alice achieves the shared secret key by computing $(g^b)^a = g^{ab}$, whereas Bob achieves it by

¹Icons in the figure are from <http://www.iconarchive.com/show/soft-icons-by-kyo-tux.html>.

computing $(g^a)^b = g^{ab}$. As a result, Alice and Bob share the secret g^{ab} , while assuming that g^{ab} can't be efficiently computed by the parameters (G, g, g^a, g^b) . This shared secret can be hashed to a symmetric key by applying a key derivation function, e.g. $\text{SHA1}(g^{ab})$.

To exemplify this protocol and the values used in it, consider the following example²:

| Alice  | Bob  |
|---|---|
| Choose $G = \mathbb{Z}_{11}^*$, $g = 2$ and $a = 3$. | |
| Compute $g^a = 2^3 = 8 \pmod{11} = 8$. | |
| Send G, g, g^a to Bob. | |
| | Choose $b = 9$. |
| | Compute $g^b = 2^9 = 512 \pmod{11} = 6$. |
| | Send g^b to Alice. |
| Compute $(g^b)^a = 6^3 = 216 \pmod{11} = 7$. | Compute $(g^a)^b = 8^9 = 134,217,728 \pmod{11} = 7$ |
| Shared secret key: $g^{ab} = 7$ | Shared secret key: $g^{ab} = 7$ |

Extended or modified versions of this basic protocol are used, for example, by the following applications (see [2],[3] and [4]):

- VPN uses the Diffie-Hellman key agreement protocol to generate a shared secret key, which is afterwards hashed to a 56 or 168-bit key to make it usable with DES or 3DES respectively.
- In SSL, various Diffie-Hellman methods are supported, for example by providing public key parameters (G, g, g^a) in a server certificate. The client simply has to inform the server about his public key parameter (g^b) , either via a certificate or a key exchange message, in order to fix a secret key between the two peers.
- The new German electronic identity card makes use of the Diffie-Hellman method as well: Running an authenticated version of the protocol between the electronic ID and a service provider, a shared secret can be exchanged. Using this secret allows generation of cryptographic keys to securely transfer personal data.

Each of these applications relies on the assumption, that the Diffie-Hellman key agreement protocol is safe. However, safety in terms of the protocol is not clearly defined. In the following, we are going to examine a few assumptions regarding the safety and analyze them.

2 The Diffie-Hellman Protocol

Before we can proceed to the actual analysis of the Diffie-Hellman protocol and the assumptions made to it, some mathematical structures have to be introduced.

2.1 Algebraic Basics - Cyclic Groups

In the previous section, we already came across the parameters G and g , as well as a, b and the computed g^a and g^b . All these are parameters of cyclic groups, the main algebraic structure used

²If you are unfamiliar with the used notation, read section 2.1 first.

by the Diffie-Hellman key exchange. The best known cyclic group is $\mathbb{Z}^+ := (\mathbb{Z}, +)$, that is the set of all integers with addition as group operation. It's called a cyclic group, because there is one element (1 or -1), called *generator*, that can generate all other elements of the whole group, by applying the group operation to itself:

$\{1^i | i \in \mathbb{Z}\} = \{(-1)^i | i \in \mathbb{Z}\} = \mathbb{Z}$ where $g^n = \overbrace{g \bullet g \bullet \dots \bullet g}^{n \text{ times}}$ for any operation \bullet .

Other examples of cyclic groups are $\mathbb{Z}_n^+ := (\mathbb{Z}_n, +_n)$ for $n \in \mathbb{N}$, where $+_n$ is the addition modulo n , e.g. $4 +_5 4 = (4 + 4) \bmod 5 = 8 \bmod 5 = 3$. \mathbb{Z}_n^+ are called finite additive groups, while \mathbb{Z}^+ is called infinite additive group. It is noteworthy that any finite cyclic group, no matter the set and operation, is isomorphic to some finite additive group, and any infinite cyclic group is isomorphic to \mathbb{Z}^+ , meaning that they can be represented in some way by either one of these. Another common cyclic group is the multiplicative group, denoted by \mathbb{Z}_n^* with multiplication modulo n as its group operation, e.g. $4 \cdot_5 4 = (4 \cdot 4) \bmod 5 = 16 \bmod 5 = 1$.

2.2 Refinements on the protocol

Being aware of cyclic groups now, we can have a deeper look at the actual key exchange protocol (Fig. 1). The parameter G Alice sends to Bob in the first step has to be a cyclic group, while $g \in G$ is a generator for that group. She chooses $a \in [1, |G|]$ and computes her public key by applying g a -times to itself: g^a . Bob (and any eavesdropper Eve) receives the parameters G, g and g^a , yet a remains unknown. By choosing a random $b \in [1, |G|]$, Bob can compute g^b and along with it $(g^a)^b = g^{ab}$, achieving the shared secret key. Bob only sends g^b to Alice, with which she can compute the shared secret key by $(g^b)^a = g^{ab}$.

Being clear about the protocol, some problems are rising:

1. How do Alice and Bob know, they are communicating with each other, and not with Eve? As the procedure doesn't include authentication, it's possible to perform a man-in-the-middle attack to extract information. This problem lies in the protocol itself and has to be taken care of when making use of it. When assuring that Alice and Bob can authenticate each other, this attack is no longer possible.
2. Assuming that authentication is not an issue, how can Alice and Bob be sure, that their secret g^{ab} is safe, when Eve knows g, g^a and g^b . Obviously group and generator have to be chosen properly, in order to grant security. Yet, what assumptions do these groups have to fulfill?

3 The Computational Diffie-Hellman assumption

Loosely speaking, the computational Diffie-Hellman assumption (CDH) states, that no efficient algorithm can compute g^{ab} from g, g^a and g^b in a certain group family \mathbb{G}^3 . A more mathematical definition is, that \mathbb{G} satisfies the (CDH) assumption, if there exists no CDH algorithm \mathcal{A} for \mathbb{G} , such that for some $\alpha > 0$ and sufficiently large n (see [1]):

$$\Pr[\mathcal{A}(\mathbb{G}, g, g^a, g^b) = g^{ab}] > \frac{1}{n^\alpha}$$

³A group family is a set of groups, that satisfies some constraint, e.g. all \mathbb{Z}_p^* , such that p is prime.

At first this might look like a sound assumption, meaning that it might be sufficient to ensure that this assumption holds for some group family \mathbb{G} , to grant security when choosing some group $G \in \mathbb{G}$ to use with the Diffie-Hellman protocol.

This is not the case, for example, when using the Diffie-Hellman protocol for the ElGamal encryption system. Alice makes the parameters (G, g, g^a) publicly available, so that encrypting a message m only requires choosing a random b , computing g^b and sending $(g^b, m \cdot g^{ab})$ to Alice. Alice decrypts by computing g^{ab} , using her private key a , and then dividing to obtain the original message m .

When choosing $G := \mathbb{Z}_p^*$ for a prime p , the CDH assumption is believed to be true, yet the system leaks information: The Legendre symbol⁴ of g^a and g^b can be easily computed. Thus, the Legendre symbol of g^{ab} is known to any attacker, and along with it the Legendre symbol of message m . Obviously, the CDH assumption alone is not sufficient to ensure security. Therefore a stronger assumption is needed to ensure the semantic security⁵ of this system.

4 The Decision Diffie-Hellman assumption

4.1 Definition

The Decision Diffie-Hellman assumption (DDH) can be described as the assumption, that, given $g, g^a, g^b \in G$, one cannot efficiently decide whether $g^c = g^{ab}$ with $a, b, c \in [1, |G|]$. Again, this can be transformed into a more mathematical definition: A group family \mathbb{G} satisfies the DDH assumption, if there is no DDH algorithm \mathcal{A} for \mathbb{G} , such that for some $\alpha > 0$ and sufficiently large n (see [1]):

$$|\Pr[\mathcal{A}(\mathbb{G}, g, g^a, g^b, g^{ab}) = \text{true}] - \Pr[\mathcal{A}(\mathbb{G}, g, g^a, g^b, g^c) = \text{true}]| > \frac{1}{n^\alpha}$$

This assumption is stronger than CDH (if one could compute g^{ab} efficiently, one could trivially decide whether $g^c = g^{ab}$ or not) and improves the security of several encryption systems, including ElGamal:

For a group G that satisfies the DDH assumption and a generator g , the system is secure under DDH, if the message space is restricted to G . This is due to the fact, that given g^a, g^b , the secret g^{ab} and along with it $m \cdot g^{ab}$ can't be distinguished from a random group element. Then no additional information about the ciphertext can be deduced from the plaintext.

4.2 Group families satisfying the DDH assumption

As we've seen before, groups have to be chosen carefully, to ensure the DDH assumption is satisfied. The following list contains a few examples of group families, for which the DDH

⁴The Legendre symbol of a (in \mathbb{Z}_p^*) is defined as

$$\left(\frac{a}{p}\right) := \begin{cases} 0 & \text{if } p|a \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic nonresidue modulo } p. \end{cases}$$

⁵By the definition of IND-CPA security, encryptions of different messages have to be indistinguishable.

assumption is believed to be true, as the best known algorithm for all of them is full discrete log, which has exponential running time (see [1]).

1. The subgroup Q_p of quadratic residues in \mathbb{Z}_p^* for some primes p and p_1 , such that $p = 2p_1 + 1$
2. The subgroup $Q_{p,q}$ of \mathbb{Z}_p^* of order q , such that $p = aq + 1$ is prime and $q > \sqrt[p]{p}$
3. The cyclic subgroup T of order $(p-1)(q-1)$ where $p, q, \frac{p-1}{2}, \frac{q-1}{2}$ are prime (note that T does not have prime order)
4. Any elliptic curve $E_{a,b}/\mathbb{F}_p$, such that p and $|E_{a,b}|$ are prime

5 Known results of DDH

5.1 Randomized Reduction

Regarding the security of DDH, it might be insightful to find an assumption that implies DDH. While it is still unknown whether CDH or any classic problem like factoring implies DDH, one can show that it is in fact implied by perfect-DDH, which is a slightly weaker assumption:

For some $\alpha > 0$ and sufficiently large n , a perfect-DDH algorithm \mathcal{A} is a polynomial time algorithm, that satisfies (see [1]):

$$\Pr[\mathcal{A}(G, g, g^a, g^b, g^c) = \text{"true"} \mid c = ab] > 1 - \frac{1}{n^\alpha}$$

$$\Pr[\mathcal{A}(G, g, g^a, g^b, g^c) = \text{"true"} \mid c \neq ab] < \frac{1}{n^\alpha}$$

Any group family G satisfies the perfect-DDH assumption, if there is no perfect-DDH algorithm for G . We are going to show, that DDH and perfect-DDH are equivalent.

Obviously the DDH assumption implies perfect-DDH, so only the converse has to be proven. Let \mathcal{O} be a DDH-algorithm (often called oracle), then it is left to show that there is a perfect-DDH algorithm \mathcal{A} that makes use of \mathcal{O} . This algorithm has to determine with overwhelming probability, whether (x, y, z) is a valid DH-triplet, for some $x, y, z \in G$, for some group G with generator g . By picking random $u, v, w \in [1, |G|]$, we construct the triplet

$$(x', y', z') = (x^w g^u, y g^v, z^w y^u x^{wv} g^{uv})$$

It is easy to show that, no matter if (x, y, z) is valid or not, the triplet (x', y', z') is indistinguishable from uniform. Moreover, (x', y', z') is always a valid DH-triplet, if (x, y, z) is valid. Thus, depending on (x, y, z) , (x', y', z') is either a uniformly random valid triplet or a completely random triplet. The perfect-DDH algorithm \mathcal{A} performs two main steps and a final step to output the result:

1. Generate k triplets (x', y', z') and query \mathcal{O} . Save the number of times the oracle answers *true* in t_1
2. Generate k random triplets in G^3 and query \mathcal{O} . Save the number of times the oracle answers *true* in t_2
3. If $|t_1 - t_2| > \epsilon k/2$ output *true*⁶

⁶Here $\epsilon \geq \frac{1}{n^\alpha}$ is the advantage of \mathcal{O} .

else output *false*

One can show that, if k is chosen, such that $k > \frac{1}{\epsilon} \log^2 \frac{1}{\delta}$, \mathcal{A} outputs the right answer with probability at least $1 - \delta$. \square

5.2 Generic algorithms

Another important point when analyzing the security of DDH is the existence of generic algorithms. A generic DDH algorithm is an algorithm \mathcal{A} , that works for all groups. If there was such an algorithm, the DDH assumption would be useless. Luckily, one can prove that the best possible generic algorithm for DDH is the best generic discrete log algorithm (Baby-Step-Giant-Step), whose runtime is $O_\epsilon(\sqrt{p})$ for a group of prime order g (see [1]).

First, let us precisely define generic algorithms⁷: A generic algorithm \mathcal{A} for \mathbb{Z}_p^+ takes as input an encoding list $(\sigma(x_1), \dots, \sigma(x_k))$, where σ is an encoding function and $x_i \in \mathbb{Z}_p^+$, and produces an output $\mathcal{A}(\sigma; x_1, \dots, x_k)$. The algorithm may query an oracle at any time (by giving it i, j and a sign bit), which will return $\sigma(x_i \pm x_j)$, depending on the algorithms query.

We will show that for a prime p , $S \subset \{0, 1\}^*$, random $a, b, c \in \mathbb{Z}_p^+$, an encoding function σ and a random bit s , any generic algorithm \mathcal{A} for \mathbb{Z}_p^+

$$\left| Pr[\mathcal{A}(\sigma; 1, a, b, w_s, w_{1-s}) = s] - \frac{1}{2} \right| < \frac{m^2}{p}$$

where $w_0 = ab$, $w_1 = c$ and m is the upper bound of oracle queries \mathcal{A} can make.

\mathcal{A} gains knowledge about the encoding $\sigma(x_i)$ of some $x_i \in \mathbb{Z}_p^+$ every time it queries the oracle. Examining the oracle's previous queries, it is possible to deduce a linear function F_i , such that $x_i = F_i(a, b, c, ab)$. If $\forall i \neq j, F_i \neq F_j : \sigma(x_i) \neq \sigma(x_j)$, then \mathcal{A} has learned the random encoding of distinct values. These values don't provide any information to \mathcal{A} , as they are independent random bit strings. So we can assume that $\exists i \neq j, F_i \neq F_j : \sigma(x_i) = \sigma(x_j)$, which might provide information to \mathcal{A} , it might learn a linear relation on a, b, c, ab . Assuming the worst case, that is \mathcal{A} can produce the correct output upon finding such values, it suffices to bound the probability that there exist i, j such that $i \neq j$ and $F_i(a, b, c, ab) = F_j(a, b, c, ab)$. Let R be this event. For $G(x, y, z) = F_i - F_j$, a polynomial of total degree 2, the probability of $(x, y, z) \in \mathbb{Z}_p^3$ being a zero of G is bounded by $2/p$ (see [1]). There are $\binom{m}{2}$ pairs of F_i, F_j , thus the total probability of R is bounded by

$$Pr[R] \leq \binom{m}{2} \cdot \frac{2}{p} < \frac{m^2}{p}$$

The algorithm produces the correct output if either R occurs or \mathcal{A} guesses the answers with probability half. \square

6 Conclusion

We have seen that the Computational Diffie-Hellman assumption isn't sufficient to ensure, that a system does not leak information. By applying the far stronger Decision Diffie-Hellman assumption, the security of such systems can be improved. The evidence from section 6.1 shows

⁷Remember that any cyclic group can be represented by the additive groups.

that the Diffie-Hellman problem cannot be decided in any non-negligible fraction of the input space, if the problem cannot be decided with overwhelming probability. Section 6.2 proves the non-existence of generic algorithms that can break DDH. Further analysis on the security of DDH can be found, for example, in [1].

Literature

- [1] D. Boneh, “The Decision Diffie-Hellman problem” (1998)
- [2] Andrew Mason, “Cisco Secure Virtual Private Networks”, Cisco Press (2001), chapter 3
- [3] William Stallings, “SSL: Foundation for Web security”,
http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html (2007)
- [4] M. Margraf, “Der elektronische Identitätsnachweis des zukünftigen Personalausweises” (2009)