Diffie-Hellman Protocol as a Symmetric Cryptosystem

Karel Burda,

Brno University of Technology, Brno, Czech Republic

Summary

The Diffie-Hellman protocol is commonly considered as an asymmetric cryptosystem. However, this paper shows that the current versions of the Diffie-Hellman protocol can, in both directions of communications, be interpreted as a series of two consecutive encryptions using a symmetric cipher. The cipher used is distinguished by a specific feature: the result of a series of several encryptions is independent of the keys' order. Each of the communicating parties encrypts the default message with its own secret key. Then, both parties send their cryptograms to each other, and each of them encrypts the received cryptogram with its own key. This way, both parties obtain the same resulting cryptogram, which is then used as a seed. The interpretation introduced above leads to a "heretical" question of whether the Diffie-Hellman protocol falls into the asymmetric or symmetric cryptosystem category.

Key words:

Diffie-Hellman protocol; public key cryptography; secret key cryptography; asymmetric cryptosystem; symmetric cryptosystem.

1. Introduction

The Diffie-Hellman protocol (abbreviated as the "DH protocol" below) has, for many decades, been used by communicating parties to generate a secret value to be shared via a public channel. The generated secret value, called the seed, is then used to derive cryptographic keys for ensuring the subsequent data transmissions.

The DH protocol was proposed by W. Diffie and M. Hellman in their breakthrough paper [1] in 1976. They characterized their protocol as a public key distribution system and classified it into a category of public key cryptography, which they also created. This type of cryptography deals with cryptosystems whose secret parameters are not shared between the communicating parties (e.g. p. 7 in [2]). Its opposite is secret key cryptography, in which the secret parameters of the cryptosystem must be shared between the parties.

The public key cryptosystems are closely related to a concept of asymmetric cryptosystems - some authors even consider these both categories to be identical. Asymmetric cryptosystems make use of a pair of keys, and the value of one of them (the so-called private key) is only known to one party, while the value of the other one (the so-called public key) is publicly known. It must be impossible to derive the value of the private key from that of the public one if such systems are to be secure (e.g. p. 544 in [3]).

The opposite of asymmetric cryptosystems are symmetric cryptosystems. In this instance, a pair of keys may again be concerned, but to determine the value of one on the basis of the knowledge of the other is possible, and values of both keys must therefore be kept secret. Both of these keys usually have the same value.

We will see below that, in the case of the DH protocol, the communicating parties do not share any secret parameters and, consequently, this protocol is included in the category of public key cryptosystems. Cryptographic parameters of the DH protocol are quite often called public and private keys (e.g. p. 48 in [4], p. 354 in [5], and p. 154 in [6]). Although, in the case of the DH protocol, some authors avoid using terms public and private keys, these authors always explicitly place the DH protocol into the category of asymmetric cryptosystems (e.g. p. 412 in [2], and p. 516 in [3]). However, as we will see in this paper, the DH protocol can also be interpreted as a symmetric cryptosystem. This ambiguous status indicates that the definitions of symmetric and asymmetric cryptosystems should be clarified.

2. DH Protocol

The DH protocol belongs to the class of cryptography based on the discrete logarithm problem (the so-called Discrete Logarithm Cryptography – DLC [4]). The mathematical basis for the contemporary versions of the DH protocol (e.g. [4], [7], [8], and [9]) is a suitable cyclic subgroup $(\boldsymbol{G}_q, \bullet)$ where $\boldsymbol{G}_q$ is a set consisting of q elements and the symbol "•" stands for the group operation. The order q of the group is a prime number, and the group operation is, typically, multiplication of integers (Finite Field Cryptography – FFC) or addition of points on an elliptic curve (Elliptic Curve Cryptography – ECC) [4].

The operation $M \bullet M \bullet \ldots \bullet M$, i.e., the group operation with e copies of the element M will be called the e-th repetition of the element M, in short written as M^e. The variable e will be called the repetition length. It holds that $M^1 = M$, $M^2 = M \bullet M$, $M^3 = M \bullet M \bullet M$, and so on. For the sake of completeness, we define $M^0 = N$ where N is the neutral (or identity) element of the group. Let us recall that, for the neutral element, equalities $M \bullet N = N \bullet M = M$ hold for each element M of the group. For the repetition of repetitions, it is evident that if $C = M^e$ then $C^d = (M e)^d = M^{e \cdot d}$.

Manuscript received July 5, 2018
Manuscript revised July 20, 2018

Let us list a few facts concerning cyclic groups (e.g. p. 9 in [10]) before describing the DH protocol itself. Every cyclic group $\boldsymbol{G}_q$ is given by its generator $G \in \boldsymbol{G}_q$, whose repetitions will generate all elements of the given group. We can thus write $\boldsymbol{G}_q = \{G^1, G^2, G^3, \ldots, G^q\}$. Another fact is that, for a cyclic group of order q, the equality $M^q = N$ holds for each $M \in \boldsymbol{G}_q$. For prime q, it is moreover true that any element except for the neutral one may be taken as a generator of $\boldsymbol{G}_q$.

The DH protocol can be described as follows with the aid of the subgroup $(\boldsymbol{G}_q, \bullet)$ with the generator G:

1. Party A chooses at random an integer a such that $1 \le a \le q-1$. This number is often called Party A's private key.
2. Now Party A calculates $A = G^a$, which is often called Party A's public key. This key is transferred by Party A to Party B via a public channel.
3. Party B choose at random an integer b such that $1 \le b \le q-1$. This number is often called Party B's private key.
4. Now Party B calculates $B = G^b$, which is often called Party B's public key. This key is transferred by Party B to Party A via a public channel.
5. Both parties are now able to calculate the shared secret element S: Party A calculates $S = B^{h\cdot a}$, and Party B calculates $S = A^{h\cdot b}$, where $h = 1$ (in the case of the FFC variant), or h is equal to the so-called cofactor (for the EEC variant). The element S is now used as the seed for deriving the keys for the subsequent data transmissions.

The following equality implies that both parties end up with the same element S:

$$A^{h\cdot b} = (G^a)^{h\cdot b} = G^{a\cdot h\cdot b} = (G^b)^{h\cdot a} = B^{h\cdot a}. \qquad (1)$$

3. Repetition Cipher

It is clear from the description of the DH protocol that, in both directions of communication, a repetition of the $C = M^e$ type is carried out twice. For the first repetition, the input is $M = G$, parameter $e = a$ or b, and output $C = A$ or B. For the second repetition, the input is $M = A$ or B, parameter $e = h\cdot b$ or $h\cdot a$, and output $C = S$. In the case of the second repetition, the equality $e = h\cdot b \bmod q$ or $h\cdot a \bmod q$ holds due to the fact that the group is cyclic. Because q is a prime number and $1 \le h \ll q$ (p. 27 in [11]), the parameter e is, in both the first and second repetitions, a random number in the range of $1 \le e \le q-1$.

Let us now view a repetition as a cipher. From the protocol definition, we know that the set of the possible values for the parameter e is the set $\boldsymbol{K} = \{1, 2, 3, \ldots, q-1\}$. The value $e = q$ is not possible because it leads to $C = G^q = N$, which is a security risk. That is why we now define $\boldsymbol{H} = \boldsymbol{G}q \setminus \{N\}$ as the set of all possible output values C for the function $C = M^e$. This set is also the set of all possible input values M; namely, if the output in the second repetition is to be $C \neq N$, the inequality $M \neq N$ must hold for the input M as well. The set $\boldsymbol{H}$ will be called the set of all possible values and $\boldsymbol{K}$ the set of all possible keys. Each of the sets $\boldsymbol{H}$ and $\boldsymbol{K}$ consists of a total of $(q-1)$ elements.

Repetition $C = M^e$, where $M, C \in \boldsymbol{H}$ and $e \in \boldsymbol{K}$, can be interpreted as a cipher $C = \mathrm{E}(M, e)$, where E stands for encryption, element M is the message, element C is the cryptogram, and repetition length e is the encryption key. To decrypt D, i.e., to determine $M = \mathrm{D}(C, d)$, we will again use repetition, this time $M = C^d$ where $d \in \boldsymbol{K}$ is the decryption key. It is true that $d = e^{-1} \bmod q$, for the decryption key, that is, $e\cdot d = (k\cdot q + 1)$, where k is a positive integer. The correctness of such decryption follows from the following chain of equalities:

$$\begin{aligned} C^d &= (M^e)^d = M^{e\cdot d} = M^{k\cdot q+1} = M^{k\cdot q}\bullet M^1 = (M^q)^k\bullet M^1 = \\ &= N^k\bullet M = N\bullet M = M \qquad (2) \end{aligned}$$

The described cipher will be called a repetition cipher.

4. Repetition Cipher Properties

We have seen that the encryption key e of a repetition cipher is, in general, different from the decryption key d. However, this cipher is not asymmetric because it is simple to calculate the decryption key with the aid of the equality $d = e^{-1} \bmod q$. The extended Euclidean algorithm, which is used to solve this equation, has the complexity of mere $O((\log_2 q)^2)$ – e.g. p. 67 in [3].

An example to illustrate a repetition cipher is shown in Fig. 1. Its basis is a multiplicative group $(\boldsymbol{G}_q, *)$ of a finite field GF(p), where $p = 11$, $q = 5$, and $G = 3$.

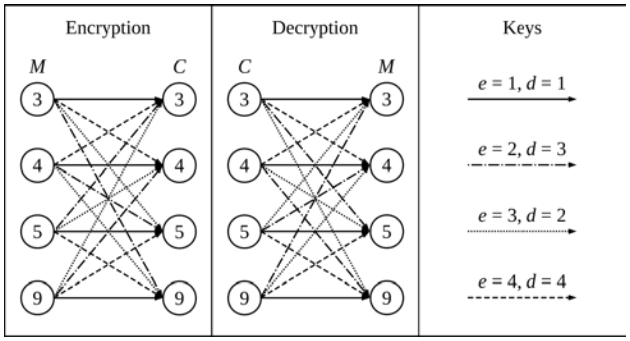

Fig. 1 Example of a repetition cipher.

The encryption or decryption function is, in this instance, $C = M^e \bmod p$ or $M = C^d \bmod p$. On the left-hand side, or in the center, the individual encryption or decryption functions are depicted for different values of the key. On the right-hand side, we can see assignments of line types to the values of the encryption key e and the corresponding values of the decryption key d. The figure clearly shows that $\boldsymbol{H} = \{3, 4, 5, 9\}$, and $\boldsymbol{K} = \{1, 2, 3, 4\}$.

Let us now study the repetition cipher in more detail. It holds that $N_M = N_C = N_K = (q-1)$, where N_M is the number of all possible messages, N_C the number of all possible cryptograms, and N_K is the number of all possible keys. We also know that any element of the subgroup $\boldsymbol{G}q$ different from the neutral element can be chosen for this subgroup's generator. In other words, each $M \in \boldsymbol{H}$ is able to generate all q elements of the set $\boldsymbol{G}q$. All $(q-1)$ different values of the key $e \in \boldsymbol{K}$ thus can, for each $M \in \boldsymbol{H}$, generate all $(q-1)$ different elements C of the set $\boldsymbol{H}$. This property implies that, for each pair (M, C), there always exists a unique key e such that $C = M^e$. If the key for each message is selected at random with a probability value of $1/N_K$ then, according to p. 49 in [12], the cipher described in this paragraph provides the perfect secrecy (i.e., the secrecy of the highest degree). Even an attacker with unlimited capacity is unable to obtain the encrypted message M from an intercepted cryptogram C without knowing the key e.

In our case, the perfect secrecy provided by the repetition cipher does not matter, because the attacker knows the encrypted message $M = G$. The goal of an attack on the DH protocol is finding the value of the encryption key a or b. If the attackers get either of these two values, they can, using the known values of B or A, calculate the secret seed S. For example, to determine the value of the key a, the attackers might utilize their knowledge of the element $A = G^a$. The element G is publicly known, and, theoretically, the value a might be obtained from A. The most efficient known algorithm to solve this problem (the so-called discrete logarithm problem, or elliptic curve discrete logarithm problem) has, however, the complexity $O(\sqrt{q})$ (e.g. p. 107 in [3], or p. 154 in [13]). This order of complexity is considered sufficient in cryptography.

5. DH Protocol as a Symmetric Cryptosystem

The DH protocol can, according to Fig. 2, be expressed as a symmetric cryptosystem; here E is the repetition cipher. Party A selects a random key a to encrypt the generator G and sends to Party B the cryptogram $A = \mathrm{E}(G, a) = G^a$. Similarly, Party B selects a random key b and encrypts the generator G using the key b. The cryptogram $B = \mathrm{E}(G, b) = G^b$ is sent to Party A. Each Party now encrypts the cryptogram obtained from the other Party using its own secret key: Party A calculates $S = \mathrm{E}(B, a) = B^a = (G^b)^a = G^{b \cdot a} = \mathrm{E}(G, b \cdot a)$, and Party B calculates $S = \mathrm{E}(A, b) = A^b = (G^a)^b = G^{a \cdot b} = \mathrm{E}(G, a \cdot b)$. Both Parties thus get an identical element S, which is the generator G, encrypted by a product of both keys' values. This result follows from a specific feature of the repetition cipher; namely, the result of two consecutive encryptions is independent of the order of the keys a and b, depending only on their product $a \cdot b$. The said feature is valid for not only a series of $n = 2$ encryptions. It is obviously also true for any $n > 1$.

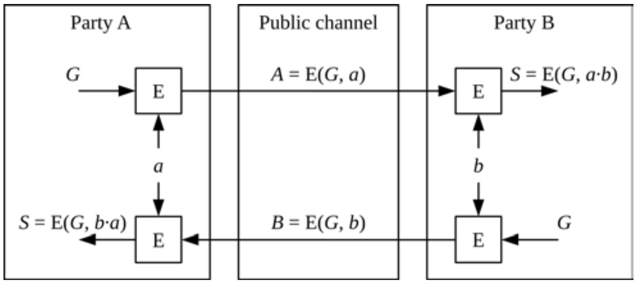

Fig. 2 The DH protocol as a symmetric cryptosystem.

The considerations presented above clearly show that the contemporary versions of the DH protocol can be described as a cryptosystem whose basic building block is a symmetric cipher. In this interpretation, parameters a and b are not private keys, but they are secret keys. Similarly, elements A and B are not public keys, but they are cryptograms where publicly known message $M = G$ is encrypted. The described interpretation classifies the DH protocol into the category of symmetric cryptosystems.

The above-mentioned conclusion is further supported by the problem of interpreting the so-called public key of the DH protocol as a cryptographic parameter. In cryptography, a key is a parameter for determination of a function (such as encrypting, decrypting, signing or verifying). Let us have a look at Fig. 2, illustrating the DH protocol: the upper branch shows two functions E; both of them represent a repetition of the input value, and the parameter (i.e., key) is a or b. On the left-hand side, the input is generator G, and the output is element A, called Party A's public key. On the right-hand side, the function E is the same, but the element A is not the parameter of this function - it is its input value. Hence it is disputable, for the DH protocol, whether the elements A and B should be called keys (i.e., parameters defining a certain function).

Another problematic point for the DH protocol is the fact that the private and public keys are not mutually interconnected. For example, in the case of an asymmetric cipher, the receiving party's private key is used to decrypt data (namely, the cryptogram) that was previously encrypted with the receiving party's public key. For a digital signature, the sender's public key is used to verify data (namely, the signature) that was previously created with the aid of the sender's private key. Regarding both these asymmetric cryptosystems, one of the keys is a parameter of a function whose input is the data on the output from a function controlled by the other member of the key pair. On the contrary, there exists no data in the DH protocol that would be affected by one of the keys and would be an input to a function defined by the other member of the key pair.

6. Comments

The DH protocol was originally defined by its authors (see [1]) on a multiplicative group of a finite field GF(p), that is, on a group with the set $\boldsymbol{G}_{p-1} = \{1, 2, 3, \ldots, p-1\}$. Here the parameter p was a prime number, the generator was a suitable number G, the repetition meant exponentiation of the generator, and the values of the private keys were subject to conditions $1 \leq a \leq p-1$ and $1 \leq b \leq p-1$. The exponential cipher on the group $\boldsymbol{G}_{p-1}$ is known as the Pohlig-Hellman cipher (e.g. p. 103 in [14]), defined by $C = M^{e} \bmod p$ and $M = C^{d} \bmod p$, while the encryption key e must be coprime with respect to (p–1) and the decryption key is $d = e^{-1} \bmod (\mathrm{p}-1)$. The selection of the set $\boldsymbol{G}_{p-1}$ in the original version of the DH protocol led to the following situation: if a and b are not coprime to (p–1), the exponentiation of the generator is not a symmetric cipher. Hence the classification of the DH protocol among asymmetric cryptosystems was quite understandable at the time of its original definition.

Further study and practical experience have shown that the original formulation of the DH protocol is vulnerable to various attacks. The most serious among those is the so-called Pohlig-Hellman decomposition [15], due to which the original group on the set $\boldsymbol{G}_{p-1}$ has been replaced with a subgroup on the set $\boldsymbol{G}_{q}$. The goal of this modification has been a natural effort to increase the security of the protocol, but in fact, it led to the hidden possibility of interpreting the DH protocol as a symmetric cryptosystem.

7. Conclusions

The contemporary versions of the DH protocol are based on a suitable subgroup of order q, where q is a prime number. This paper shows that in such groups the group operation involving e copies of element M, i.e., repetition $C = M \bullet M \bullet \ldots \bullet M = M^{e}$, can be interpreted as a symmetric cipher in which the resulting element C is the cryptogram, the element M is the message, and the repetition length e with $1 \leq e \leq q-1$ is the encryption key. The group operation used in the DH protocol must, of course, ensure that the key should be protected, so that it must be impossible to determine the key e based on the knowledge of C and M.

Since $(M^{a})^{b} = M^{a \cdot b} = (M^{b})^{a}$, the result of a series of two encryptions with keys a and b is independent of the order of the keys. This fact is the basis of the DH protocol. Each of the communicating parties encrypts the default message M with its own key to the cryptogram M^{a} or M^{b}, send them to each other, and each of them will encrypt the received cryptogram with its own key. Both parties as such obtain the same resulting cryptogram $M^{a \cdot b}$, which is then used as the seed.

Our finding that the DH protocol can also be interpreted as a symmetric cryptosystem implies that the definitions of symmetric and asymmetric cryptosystems should be clarified.

References

[1] Diffie W., Hellman M.: New directions in cryptography. IEEE Transactions on Information Theory, 1976, Vol. 22, No. 6, p. 644-654.

[2] Oppliger R.: Contemporary Cryptography. Artech House, Norwood 2005.

[3] Menezes A. J., Oorschot P. C., Vanstone S. A.: Handbook of Applied Cryptography. CRC Press, Boca Raton 1996.

[4] Barker E. aj.: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A, Revision 3. NIST, Gaithersburg 2018.

[5] Keith M. M: Everyday cryptography. Oxford University Press, Oxford 2017.

[6] Henk C. A. van Tilborg: Encyclopedia of Cryptography and Security. Springer, N. York 2005.

[7] American National Standard X9.42-2003 (R2013), Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography. ABA, Washington 2003.

[8] American National Standard X9.63-2011 (R2017), Key Cryptography for the Financial Services Industry: Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. ABA, Washington 2011.

[9] SEC 2: Recommended Elliptic Curve Domain Parameters. Certicom, Mississauga 2010.

[10] Polites G. W.: An Introduction to the Theory of Groups. International textbook company, Scranton 1968.

[11] Digital Signature Standard (DSS). FIPS PUB 186-4. NIST, Gaithersburg 2013.

[12] Stinson D. R.: Cryptography - Theory and Practice. CRC Press, Boca Raton 1995.

[13] Hankerson D., Menezes A., Vanstone S.: Guide to Elliptic Curve Cryptography. Springer, New York 2004.

[14] Denning D. E.: Cryptography and Data Security. Addison-Wesley, Reading 1982.

[15] Pohlig S. C., Hellman M. E.: An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Transactions on Information Theory, 1978, Vol. 24, No. 1, p. 106-110.

Karel Burda received the M.S. and Ph.D. degrees in Electrical Engineering from the Liptovsky Mikulas Military Academy in 1981 and 1988, respectively. During 1988-2004, he was a lecturer in two military academies. At present, he works at Brno University of Technology. His current research interests include the security of information systems and cryptology.