SECURITY IMPROVMENTS TO THE DIFFIE-HELLMAN SCHEMES

Malek Jakob Kakish
Amman Arab University, Department of Computer Information Systems,
P.O.Box 2234, Amman 11953, Jordan.
Email: doctor_malek@yahoo.com; malek@aau.edu.jo

ABSTRACT

The Diffie-Hellman key agreement protocol and the Diffie-Hellman encryption/decryption cryptosystem are historical the firstly defined Public key cryptosystems that enables communicating parties over unsecure communication channel to agree upon a shared secret key and to encrypt /decrypt messages.

In praxis Diffie-Hellman key agreement is very often used as part of security protocols or security standards to secure data over public and communication systems, thus the security of the Diffie-Hellman is critical because any weaknesses can lead such systems to become vulnerable against attacks.

This paper introduces a security improvement that makes the Diffie-Hellman key agreement and encryption scheme more secure against attacks, such as the known plaintext attacks, it suggests the use of randomized parameter in both schemes, this will allows to produce a new shared secret key each time a communication session is build and to generate different encryption messages for all kinds of messages even for same message, thus making the Diffie-Hellman more secure compared with the basic version of the Diffie-Hellman.

Keywords: *Diffie-Hellman problem, Diffie-Hellman encryption, Discrete Logarithm Problem, Public Key Cryptosystems, Private Key Cryptography, Crypto Analysis.*

1. INTRODUCTION

The computer and communication technologies are one of the most rising technologies, they build the backbone for the economy thus it is important to have suitable security technologies and systems that fulfill the security needs and requirements for these technologies.

Many security systems and algorithms have been developed that are specified in standards, such standards comes mostly from a well known standard organizations (e.g. Internet Architecture Board (IAB), Internet Engineering Task Force (IETF), etc.) that specify a huge set of security protocols, algorithms and applications which provide security services and meets our needs for data privacy and secure communication.

A major rule in these specifications plays the Cryptography. Cryptography underlies many of the security mechanisms and it builds the science of data encryption and decryption. Cryptography enables us to securely store sensitive information or transmit them across insecure networks such that it cannot be read by anyone except the intended recipient. By using encryption we gain privacy, authenticity, integrity, and limited access to data. In Cryptography we differentiate between private key cryptographic systems (also known as conventional cryptography systems) and public key cryptographic systems.

Private Key Cryptography, also known as secret-key also known as symmetric-key encryption, has an old history [1], and is based on using one shared secret key for encryption and decryption, managing such keys can work for few communicating parties but is impractical for large increasing set of communication parties. The development of new computer and communication technologies helped to define many modern private key cryptographic systems, in the 1960's many cryptosystems were defined which are based on Feistel cipher, e.g. Data Encryption Standard (DES), Triple Data Encryption standards (3DES), Advanced Encryption Standard (AES), The International Data Encryption Algorithm (IDEA), Blowfish, RC5, CAST, etc.[2]

A new concept in cryptography was introduced in 1976 by Diffie and Hellman [3], this new concept was called public-key cryptography and is based on using two keys (Public and Private key). The use of public key cryptography solved many weaknesses and problems in private key cryptography, many public key cryptographic systems were specified (e.g. RSA [4], ElGamal [5], Diffie-Hellman key exchange [3], elliptic curves [6], etc.). The security of such Public key cryptosystems is often based on apparent difficulties of some mathematical number theory problems (also called “trap door, one way functions") e.g. the Discrete Logarithm Problem over finite fields, the Discrete Logarithm Problem on Elliptic Curves, the Integer Factorization Problem or the Diffie-Hellman Problem, etc. [2].

One of the firstly defined and often used to securely exchange keys is the Diffie-Hellman key agreement protocol and is often an essential part of authentication protocols or part of a whole security system, e.g. Diffie-Hellman is used in Internet security standard protocols IPSEC to secure transmitted data through public networks and is mostly used in communication systems today. The Diffie-Hellman protocol is also called key exchange protocol and has

been patented in 29 of April 1980, in the USA under Patent 4,200,770 (in 6 Sep. 1997 expired), it is assigned to Stanford University and covers the Diffie Hellman key agreement and mention three persons, Hellman, Diffie, and Merkle as inventors [2].
Many well known standard organizations specified security standards which define the implementation and the use of Diffie-Hellman encryption/decryption or key agreement protocol [7] [8].
Due to the widely use of the Diffie-Hellman key agreement protocol and the encryption scheme, is it critical to ensure a high level of security for the Diffie-Hellman schemes, in this paper I introduce a new enhancement to the security of the Diffie-Hellman schemes, this is achieved by using randomized parameter in the key agreement protocol and the encryption process, the use of randomized parameters will make it more difficult for an attacker or cryptanalysis people to break the Diffie-Hellman schemes.

2. PROBLEM FORMULATION

The security of many cryptosystems and protocols are based on the Diffie-Hellman problem, this means that if in the future the Diffie-Hellman problem is efficiently solved then many of these cryptosystems will become insecure, e.g. the El-Gamal cryptosystem (a variant of Diffie-Hellman scheme) is a well known public key cryptosystem, his security depends on the intractability of the Discrete Logarithm Problem as well as the Diffie Hellman problem.
The Diffie-Hellman schemes considered in this paper use Z_p^* groups of prime order p, but in literature we find also that Diffie-Hellman can also be considered on groups of composite integers n [2].
Let p be a prime number, then Z_p denotes the set of integers $\{0, 1, 2, \dots, p-1\}$, where addition and multiplication are performed modulo p. It is well-known that there exists a non-zero element $g \in Z_p$ such that each non-zero element in Z_p can be written as a power of g such element g is called a generator of Z_p. A group is called cyclic if such element g exists.

Definition A Field is a non empty set F of elements with two operations "+" (called addition) and " ・ " (called multiplication) satisfying the following axioms: for all $a, b, c \in F$,

i. F is closed under + and ・, i.e., $a + b$ and $a ・ b$ are in F;
ii. Commutative laws: $a + b = b + a$, $a ・ b = b ・ a$;
iii. Associative laws: $(a + b) + c = a + (b + c)$, $a ・ (b ・ c) = (a ・ b) ・ c$;
iv. Distributive law: $a ・ (b + c) = a ・ b + a ・ c$.

Furthermore, two distinct identity elements 0 and 1 (called the additive and multiplicative identities, respectively) must exist in F satisfying:

v. $a + 0 = a$ for all $a \in F$;
vi. $a ・ 1 = a$ and $a ・ 0 = 0$ for all $a \in F$;
vii. For any a in F, there exists an additive inverse element $(-a)$ in F such that $a + (-a) = 0$;
viii. For any $a \neq 0$ in F, there exists a multiplicative inverse element a^{-1} in F such that $a ・ a^{-1} = 1$

Definition A Finite field of prime order p or prime power $q = p^f$ $(f >= 1)$ is commonly denoted F_q or $GF(q)$ (Galois field) and because Z_m is a field if and only if m is a prime, we denote the field Z_m by F_m. This is called a prime field.

Definition For $n \geq 1$, let $\varphi(n)$ denote the number of integers in the interval $[1,n]$ which are relatively prime to n. The function φ is called the Euler phi function (or the Euler totient function)

Definition Let $\alpha \in Z_p^*$ .If the order of α is $\varphi(n)$, then α is said to be a generator or a primitive element of Z_p^*. If Z_p^* has a generator, then Z_p^* is said to be cyclic.

Definition The Diffie-Hellman Problem is the following: given a prime p, a generator α of Z_p^*, and elements $\alpha^a \bmod p$ and $\alpha^b \bmod p$, find $\alpha^{ab} \bmod p$.

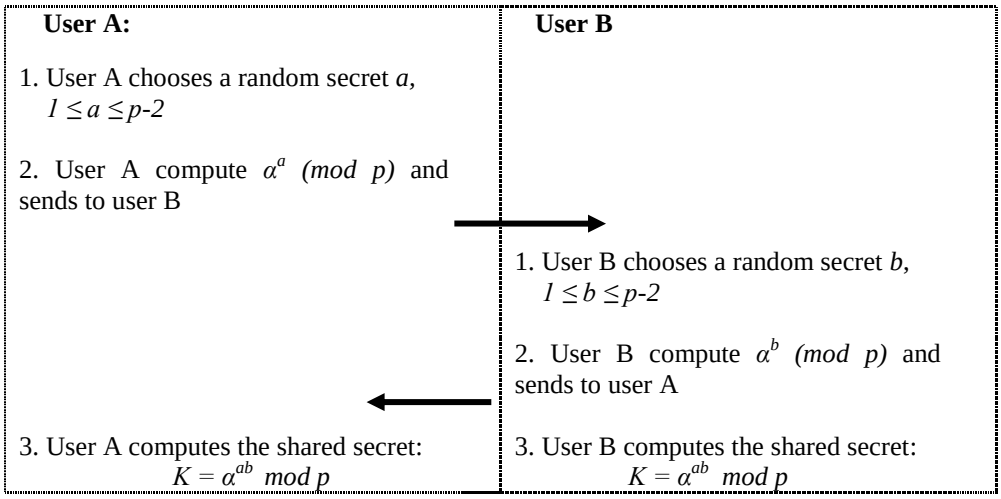
Definition The Discrete Logarithm Problem is the following: given a prime p, a generator α of Z_p^*, and an element $\beta \in Z_p^*$, find the integer x, $0 \leq x \leq p-2$, such that $\alpha^x = \beta \pmod p$.

From the above definition we can easily conclude that if the Discrete Logarithm Problem over finite fields is efficiently solved then the Diffie-Hellman problem is broken because if we calculate a or b from publicly known α^a or α^b then we can easily compute the shared secret α^{ab}, for more information see [9]
The Diffie-Hellman problem has been studied for many years but still an efficient solution was not found thus it is considered as being difficult if the parameters are suitably chosen, but if the factors of $p-1$ are known or are small

integers then the Diffie-Hellman problem can be easily solved using e.g. Index Calculus method, Pohlig-Hellman algorithm, etc., for more information see [2].
This implies that when generating Diffie-Hellman keys, it is important that the prime factors of *p-1* should be selected in sufficient size such that factoring *p-1* should be computationally infeasible.
Now I describe the basic version of the Diffie-Hellman key agreement protocol which is widely used in security protocols and which allows communicating parties that have never met before to establish a secret shared key over an open channel, this shared key can also be used to encrypt data between the two communicating parties.

Protocol: Diffie-Hellman key agreement (basic version)
RESULT: a shared secret *K* known to both communicating parties A and B.

1. One-time setup: An appropriate prime *p* and a generator α of Z_p^* $(2 \le \alpha \le p-2)$ are selected and published.
2. Protocol messages: Each time a shared key is required parties:



Here we should also note that the Diffie-Hellman key agreement protocol does not authenticate the protocol users, this means that an adversary can perform intercepting, modifying or injecting of messages without being discovered, a list of attacks on Diffie-Hellman protocol can be found in [2].

Algorithm: The Diffie-Hellman encryption/decryption scheme (basic version):
User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:
(a) Obtain user A authentic public key α^a.
(b) Represent the message as an integer *m* in the interval *[0,p-2]*
(c) Use private key *b* and compute shared secret α^{ab} *(mod p)*
(d) Compute $c = m*\alpha^{ab} (mod\ p)$
(e) Send the encrypted text message c to user A .

2. Decryption. To recover plaintext m from c, user A should do the following:
(a) Obtain user B authentic public key α^b.
(b) Use private key *a* and compute shared secret α^{ab} *(mod p)*
(c) Compute inverse element $\alpha^{-ab} (mod\ p)$
(d) Recover $m = m * \alpha^{ab} * \alpha^{-ab} \ mod\ p$

The Diffie-Hellman key agreement protocol provide the secrecy of the shared key because only the communicating parties knows *a* and *b*, thus only they can compute the shared secret key, on the other hand the problem rise in that neither one of the communicating parties is assured of the identity of the other (man-in-the-middle attack), this problem can be solved if both parties have access to certifications that binds their identity with the corresponding public key or if they use a public parameter distribution over trusted channels.
An authenticated Diffie-Hellman key agremment protocol, also called Station-to–Station (STS) protocol, was described by Diffie, van Oorschot, and Wiener in 1992 [10] that provide protection against man-in-the-middle attack this is achieved by using digital signatures and public key certificates.
The above basic version of the Diffie-Hellman encryption, decryption and key agreement protocol does not contain any randomized parameter, thus a repeated message block will always lead to the same encryption block, and furthermore a chosen plain text attack can be performed.

3. THE SECURITY IMPROVEMENTS OF THE DIFFIE-HELLMAN

The following protocol describes the modified Diffie-Hellman key agreement protocol over unsecure communication channel.

Protocol: Diffie-Hellman key agreement (modified version)
RESULT: a shared secret *S* known to both communicating parties A and B.

1. One-time setup: An appropriate prime p and a generator α of Z_p^* $(2 \le \alpha \le p-2)$ are selected and published.
2. Protocol messages: Each time a shared key is required parties:

User A	**User B**
1. User A chooses a random secret a, $1 \le a \le p-2$	
2. User A chooses a random integer k, $1 \le k \le p-2$	
3. User A build $\alpha^{ak} \pmod p$ and sends (α^{ak}) to user B →	
	1. User B chooses a random secret b, $1 \le b \le p-2$
	2. User B builds $\alpha^b \pmod p$ and sends to user A ←
4. User A computes the shared secret $S = \alpha^{abk} \bmod p$	3. User B computes the shared secret $S = \alpha^{abk} \bmod p$

In the modified Diffie-Hellman key agreement protocol above the shared secret key will always look different even though we are using the same public keys (α^a, α^b) , this is due to the randomized parameter k. Using this randomized parameter will assure more security.

Example: (Diffie-Hellman Key agreement protocol)
Assume public parameter $p = 2357$, and a generator $\alpha = 2$ of Z_p^*

User A	**User B**
1. User A chooses a random secret $a = 306$	
2.User A chooses a random integer $k = 767$	
3. User A compute $\alpha^{ak} = 1212 \pmod{2357}$ and send (1212) to user B →	
	1.User B chooses a random secret $b = 658$
	2. User B compute $\alpha^b = 1621 \pmod p$ and sends to user A ←
4. User A computes the shared secret $S = \alpha^{abk} = 1849 \bmod 2357$	3. User B computes the shared secret $S = \alpha^{abk} = 1849 \bmod 2357$

The following algorithm describes the modified Diffie-Hellman encryption/ decryption.

Algorithm: The modified Diffie-Hellman encryption/decrption scheme:
User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:
(1.1) Obtain user A authentic public key α^a.
(1.2) Represent the message as an integer *m* in the interval *[0, p-2]*
(1.3) Choose a random integer *k, 1* $\leq k \leq$ *p-2*
(1.4) Use B private key *b* and Compute α^{ab} *(mod p)*
(1.5) Compute $c = m*\alpha^{abk}$ *(mod p)*
(1.6) Compute $k\alpha^{ab}$ *(mod p)*
(1.7) Send the encrypted text message $(c, k\alpha^{ab})$ to user A.

2. Decryption. To recover plaintext *m* from *c*, user A should do the following:
(2.1) Obtain user B authentic public key α^b.
(2.2) Use A private key *a* and Compute α^{ab} *(mod p)*
(2.3) Compute α^{-ab} *(mod p)*
(2.4) Recover *k*, compute $\alpha^{-ab} * k\alpha^{ab} = k \bmod p$
(2.5) Compute $\alpha^{-abk} \bmod p$
(2.6) Recover *m*, compute $\alpha^{-abk} * m*\alpha^{abk} = m$ *(mod p)*

Example: (Diffie-Hellman Ecryption/Decryption)
Key Generation: Assume *p = 2357*, generator $\alpha = 2$,

1. Encryption. User B should do the following:
(1.1) Obtain user A authentic public key $\alpha^a = 2082 \bmod 2357$
(1.2) message *m = 900*
(1.3) Random integer *k = 767*
(1.4) B private key *b = 658*, Compute $(2082)^{658} = 2030 \pmod{2357}$
(1.5) Compute $900 * 2030^{767} = 58 \pmod{2357}$
(1.6) Compute $767*2030 = 1390 \bmod 2357$
(1.7) Send the encrypted text message *(58, 1390)* to user A.

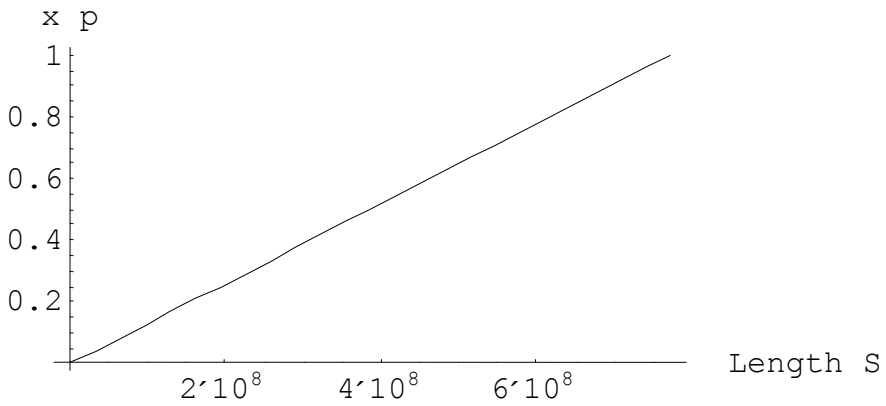
2. Decryption. To recover plaintext *m* from *c*, user A should do the following:
(2.1) Obtain user B authentic public key $\alpha^b = 1621 \bmod 2357$.
(2.2) A private key *a =306*, Compute $\alpha^{ab} = (1621)^{306} = 2030 \pmod{2357}$
(2.3) Compute $(2030)^{-1} = 728 \bmod 2357$
(2.4) Recover k, compute $1390*728 = 767 \bmod 2357$
(2.5) Compute $728^{767} = 747 \bmod 2357$
(2.6) Recover *m*, $747 * 58 = 900 \bmod 2357$

Diffie-Hellman basic version is vulnerable against known plain-text attack; a known-plaintext attack is one where the adversary has a quantity of plaintext and corresponding ciphertext [2].

Given such a sorted set $S = \{\{p_1,c_1\},\{p_2,c_2\},\ldots,\{p_n,c_n\}\}$ (where $p_i \in P$ plaintext set, $c_i \in C$ ciphertext set, $n <= p$, p is the order of Z_p^*) an adversary can determine the plaintext p_x if the corresponding c_x is in S.

The following example shows the relationship between the length of S and the probability function $f = x/p$ of finding a searched element p_x in *S*.

Example: assume *p = 771143167*, the function $f = x/p$, where $x \in Z_p^*$



The optimal case is when $n = p$, this will allow us to determine p_i for each given c_i but this also means that we will need a huge storage space of *2p* elements which is impractical for a *1024* bits *p*, and it would also require *O(p*log(p))* comparison operations to sort list *S*.

The modified version of Diffie Hellman encryption algorithm described above use *k* as randomizing parameter; this will protect the encrypted text against known plain text attacks described above, because even if we know p_x, in the equation:

$$p_x = k_x m_x$$

k_x and m_x will still remain unknown.

Important for Diffie-Hellman key agreement and encryption schemes regarding security consideration, is the size of the modulus *p* where *p-1* prime factors should be so selected sufficiently large such that factoring is computationally infeasible. For a moderate security level should be at least *2048* bits length. Furthermore it is recommended that *p* should be randomly chosen and not of some special case binary bit structure to avoid attacks such as the exhaustive search attack.
Some of the attacks of the Diffie-Hellman schemes are attacks on the implementation [11] [12] other powerful attacks on the Diffie-Hellman schemes are attacks on the integer factorization problem; e.g. the elliptic curve factoring algorithm [13], quadratic sieve [14] and number field sieve [15].
In 2010, the largest number factored by a general-purpose factoring algorithm was *768* bits long [16] using distributed implementation thus some experts believe that *1024*-bit keys may become breakable in the near future so it is currently recommended to use *4096*-bit keys for long term security.

For more information about attacks on the Diffie-Hellman schemes see [2].

4. CONCLUSIONS

In this paper I briefly discussed improving the security of the Diffie-Hellman key agreement and encryption decryption schemes, these improvements use randomized parameter to secure every shared secret key and every encrypted message block so that even if the same message is sent many times the encrypted message block will look different.
The major advantage gained here is that the security improvement described in this paper protects Diffie-Hellman schemes from a known plaintext attack, thus making the Diffie-Hellman schemes semantically secure, this is important because as mention in the introduction above, the Diffie-Hellman is implemented in many internet security standards and protocol and a weak Diffie-Hellman can make the whole system compromised. One solution that is used in praxis to overcome this problem is the use of padding bits in the generation process of the keys or in the encryption decryption process, but this may not always works if the adversary knows the padding bits.

This security improvements described in this paper provides makes the Diffie-Hellman schemes more immune against adversary attacks nevertheless it should be noted that the Diffie-Hellman p modulus bit length should be at least 2048 to ensure a moderate security and to avoid powerful attacks on the discrete logarithm problem. This security consideration and other mentioned in literature should be used to define an improved version of the Diffie-Hellman schemes and also signature scheme [17].

The security of many public key cryptographic systems such as the Diffie-Hellman crypto system is based the intractability of the Diffie Hellman Problem and the Discrete Logarithm Problem, this may change in the future due to new mathematic insights or new computer technologies [18] that may allow us to solve this problem in short time.

5. REFERENCES

[1]. Kahn D., 1967. The Code breakers: The comprehensive History of Secret Communication from Ancient to the Internet

[2]. Menezes A., et. al., 1999. Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7

[3]. Diffie W., Hellman M., 1976. "New directions in cryptography", IEEE Transactions on Information Theory, volume 22, pages 644-654.

[4]. Rivest R. L., et al, 1978. A method for obtaining digital signatures and public key cryptosystems. Commun. of the ACM, volume 21, pages 120-126.

[5]. ElGamal T., 1985. "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, volume 31, pages 469-472.

[6]. Koblitz N., 1987. Elliptic curve cryptosystems. Mathematics of Computation, volume 48, pages 203-209.

[7]. ANSI press, 1999. Specified X9.42: key management using Diffie-Hellman, this standard specifies several variations of unauthenticated Diffie-Hellman key agreement, providing shared symmetric keys.

[8]. IEEE press, 2000. Specified P1363: Standard for RSA, Diffie-Hellman and related public-key cryptography, which includes specifications for elliptic curve systems.

[9]. Maurer U.M., 1994. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, 271–281.

[10]. Diffie, W.; van Oorschot, P. C.; Wiener, M. J. (1992), "Authentication and Authenticated Key Exchanges", Designs, Codes and Cryptography (Kluwer Academic Publishers) 2: 107–125,

[11]. Kocher P., 1996. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems, CRYPTO '96 Proceedings of the 16th Annual.

[12]. Francois J., Raymond A., 1998. Security Issues in the Diffie-Hellman Key Agreement Protocol, IEEE Trans. on Information Theory, pages 1–17.

[13]. Dixon B., , Lenstra A.K., 1993. Massively parallel elliptic curve factoring, Advances in Cryptology, Eurocrypt '92, Lecture Notes in Comput. Sci. 658, pp.183--193.

[14]. Pomerance C., 1984. The quadratic sieve factoring algorithm, In EUROCRYPT 169-182.

[15]. Buchmann J., et. al., 1993. An implementation of the general number field sieve, In Advances in Cryptology - Crypto, pages 159-166, Springer-Verlag.

[16]. RSA Laboratories, 1991 .the RSA Factoring Challenge http://www.rsa.com/rsalabs/node.asp?id=2092

[17]. Mihir A. M., et al., 2001. DHIES: An encryption scheme based on the Diffie-Hellman Problem, Topics in Cryptology – CT-RSA 2001, volume 2020 of Lecture Notes in Computer Science, pages 143–158. Springer-Verlag, Berlin Germany.

[18]. Shor P. W., 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26, 1484–1509.