ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

Elliptic Curve Diffie-Hellman (ECDH)

Ajit karki
MTech. (Computer Science and Engineering).
Assistant professor.Deptt. Of Computer Science.
The ICFAI University, Sikkim, India
Ranka Road sichey, Gangtok,Sikkim.
ajitkarki4@gmail.com

Abstract:

Cryptography is the technique of hiding a message in some unintelligible format so that the message lies hidden in plain sight of an unintended person. The techniques of cryptography are centuries old. With technological advancement, techniques have evolved significantly. Public key cryptography offers a wide range of security over the various modes of transferring data, especially over Internet. The security of a public key encryption is stronger only if the authenticity of the public key is ensured. Data encryption standards like RSA and Diffie- Hellman are becoming incapable due to requirement of large number of bits for cryptographic process. As of latest, ECC (Elliptic Curve Cryptography) has become the latest trend in the cryptographic scenario. This paper presents the implementation of ECC for encryption/decryption and authentication process. It is worth noting that brute force attack on ECC is infeasible due to the discrete logarithm problem it possesses. Elliptic Curve Diffie Hellman (ECDH) is an Elliptic Curve variant of the standard Diffie Hellman algorithm.

Keywords: Encryption, Decryption, Elliptic Curve cryptography, Encoding, Decoding.

Introduction: Overview of Methods:

Diffie-Hellman key agreement requires that both the sender and recipient of a message have key pairs. By combining one's private key and the other party's public key, both parties can compute the same shared secret number. This number can then be converted into cryptographic keying material. That keying material is typically used as a key-encryption key (KEK) to encrypt (wrap) a content- encryption key (CEK) which is in turn used to encrypt the message data.

Elliptic Curve Diffie Hellman
(ECDH) is used for the purposes of key agreement. Suppose two people, Alice and Bob, wish to exchange a secret key with each other. Alice will generate a private key d_A and a public key $Q_A=d_AG$ (where G is the generator for the curve). Similarly Bob has his private key d_B and a public key $Q_B=d_BG$. If Bob sends his public key to Alice then she can calculate $d_AQ_B=d_Ad_BG$. Similarly if Alice sends her public key to Bob, then he can calculate $d_bQ_A=d_Ad_BG$. The shared secret is the x co-ordinate of the calculated point d_Ad_BG. Any eavesdropper would only know Q_A and Q_B, and would be unable to calculate the shared secret.

Background:

The Diffie-Hellman algorithm, introduced by Whitfield Diffie and Martin Hellman in 1976, was the first system to utilize “public-key” or “asymmetric” cryptographic keys. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Over the past 30 years, public key cryptography has become a mainstay for secure communications over the Internet and throughout many other forms of communications. It provides the foundation for both key management and digital signatures. In key management, public key cryptography is used to distribute the secret keys used in other cryptographic algorithms (e.g. DES). For digital signatures, public key cryptography is used to authenticate the origin of data and protect the integrity of that data.

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

For the past 20 years, Internet communications have been secured by the first generation of public key cryptographic algorithms developed in the mid-1970's. Notably, they form the basis for key management and authentication for IP encryption (IKE/IPSEC), web traffic (SSL/TLS) and secure electronic mail.
In their day these public key techniques revolutionized cryptography. Over the last twenty years however, new techniques have been developed which offer both better performance and higher security than these first generation public key techniques. The best assured group of new public key techniques is built on the arithmetic of elliptic curves. This paper will outline a case for moving to elliptic curves as a foundation for future Internet security. This case will be based on both the relative security offered by elliptic curves and first generation public key systems and the relative performance of these algorithms. While at current security levels elliptic curves do not offer significant benefits over existing public key algorithms, as one scales security upwards over time to meet the evolving threat posed by eavesdroppers and hackers with access to greater computing resources, elliptic curves begin to offer dramatic savings over the old, first generation techniques. The two noteworthy first generation public key algorithms used to secure the Internet today are known as RSA and Diffie-Hellman (DH). The security of the first is based on the difficulty of factoring the product of two large primes. The second is related to a problem known as the discrete logarithm problem for finite groups. Both are based on the use of elementary number theory. Interestingly, the security of the two schemes, though formulated differently, is closely related.

Both algorithms have been subject to intense scrutiny since their invention around 1975. In the years immediately following their invention, there were a number of attacks based on a variety of degenerate ways to generate the prime numbers and such that define the system. Such parameters were quickly excluded from specifications. In the public domain, more general theoretic attacks on the fundamental problems of factoring and discrete logs made steady progress until the early 1990's. Since that time, no dramatic improvements in these attack algorithms have been published. However, there have been several efforts aimed at designing theoretical special purpose computers that would implement the existing attack algorithms far faster than general computing resources.
In 1985, Neal Koblitz and Victor Miller independently suggested the use of elliptic curves in public key cryptography. Supporters of elliptic curve cryptography (ECC) claim that ECC requires much smaller keys than those used in conventional public key cryptosystems, while maintaining an equal level of security. The use of elliptic curves therefore allows faster encryption and decryption. Since their use in cryptography was discovered in 1985, elliptic curve cryptography has also been an active area of study in academia. Similar to both RSA and Diffie-Hellman, the first years of analysis yielded some degenerate cases for elliptic curve parameters that one should avoid. However, unlike the RSA and Diffie-Hellman cryptosystems that slowly succumbed to increasingly strong attack algorithms, elliptic curve cryptography has remained at its full strength since it was first presented in 1985.

Diffie-Hellman Key Exchange

A key agreement scheme is a procedure by which two or more parties agree upon a value from which they can subsequently derive one or more keys for use in a symmetric encryption and/or data authentication scheme. Neither party completely determines the key value on their own. Instead, they both contribute to the final key value. And, most important, anyone who observes the exchanges between the two parties cannot tell what the final result will be. The cryptosystem we aim to achieve is one where the sender and receiver exchange pieces of information via an insecure network resulting in both parties sharing a common secret whereas anyone else, who intercepts the transfer of the message, is unable to discover the shared secret. This shared secret is what is used as a key in conventional cryptosystems. Such a
system is a full public key cryptosystem. We illustrate this key exchange protocol with an example:
Alice and Bob aim to exchange information using a public key cryptosystem.
They publicly choose a cyclic group G and a generator x of G.

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

Alice and Bob choose private keys **a** and **b** respectively, where a and b are random integers.

Alice computes x^a, Bob computes x^b and they exchange these values over an insecure network.
On receiving the information from each other, both Alice and Bob compute the

a value x^{ab} using their private keys and the fact that $x^{ab} = (x^a)^b = (x^b)$. Now, both Alice and Bob share a secret, namely, the value x^{ab}. That is, Alice and Bob have exchanged a key, x^{ab}, that can now be used in a conventional cryptosystem to encrypt any messages between Alice and Bob. If the message was intercepted, the eavesdropper, in order to decipher the message, has to obtain the value x^{ab} from x, x^a and x^b. This problem is called the Diffie-Hellman problem. One way to tackle this problem is to try to compute a from x^a. This is known as the discrete logarithm problem. With the basics of public key cryptography in hand, we are now in a position to apply elliptic curves to public key cryptography in order to generate public and private keys.

Explanation including encryption mathematics

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo *p*, where *p* is prime and *g* is primitive root mod *p*. Here is an example of the protocol, with non-secret values in **blue** and secret values in **red**. Small integers are used for clarity, but actual implementations require using
much larger numbers to achieve security.

Alice				**Bob**		
Secret	Public	Calculates	Sends	Calculates	Public	Secret
a	p, g		$p,g \rightarrow$			b
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, s

Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$.

Alice chooses a secret integer $a = 6$, then sends Bob
$A = g^a \bmod p$
$A = 5^6 \bmod 23$
$A = 15{,}625 \bmod 23$
$A = 8$
Bob chooses a secret integer $b = 15$, then sends Alice
$B = g^b \bmod p$
$B = 5^{15} \bmod 23$
$B = 30{,}517{,}578{,}125 \bmod 23$
$B = 19$
Alice computes $s = B^a \bmod p$
$s = 19^6 \bmod 23$
$s = 47{,}045{,}881 \bmod 23$
$s = 2$
Bob computes $s = A^b \bmod p$
$s = 8^{15} \bmod 23$
$s = 35{,}184{,}372{,}088{,}832 \bmod 23$
$s = 2$
Alice and Bob now share a secret (the number 2) because 6×15 is the same as 15×6.
Both Alice and Bob have arrived at the same value, because $(g^a)^b$ and $(g^b)^a$ are equal mod *p*. Note that only *a*, *b*, and *($g^{ab} \bmod p = g^{ba} \bmod p$)* are kept secret. All the other values – *p*, *g*, $g^a \bmod p$, and $g^b \bmod p$ – are sent in the clear. Once Alice and Bob compute the hared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of *a*, *b*, and *p* would be needed to make this example secure, since there are only 23 possible results of *n* mod 23. However, if *p* is a prime of at least 300 digits, and *a* and *b* are at least 100 digits

long, then even the fastest modern computers cannot find *a* given only *g*, *p*, $g^b \bmod p$ and $g^a \bmod p$. The problem such a computer needs to solve is called the discrete logarithm problem.

Elliptic Curve Security and Efficiency:

The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman. The US National Institute for Standards and Technology has recommended that these 1024-bit systems are sufficient for use until 2010. After that, NIST recommends that they be upgraded to something providing more security. The question is what should these systems be changed to? One option is to simply increase the public key parameter size to

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

a level appropriate for another decade of use. Another option is to take advantage of the past 30 years of public key research and analysis and move from first generation public key algorithms and on to elliptic curves. One way judgments are made about the correct key size for a public key system is to look at the strength of the conventional (symmetric) encryption algorithms that the public key algorithm will be used to key or authenticate. Examples of these conventional algorithms are the Data Encryption Standard (DES) created in 1975 and the Advanced Encryption Standard (AES) now a new standard. The length of a key, in bits, for a conventional encryption algorithm is a common measure of security. To attack an algorithm with a k-bit key it will generally require roughly 2k-1 operations. Hence, to secure a public key system one would generally want to use parameters that require at least 2k-1 operations to attack. The following table gives the key sizes recommended by the National Institute of Standards and Technology to protect keys used in conventional encryption algorithms like the (DES) and (AES) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security.

Symmetric Key Size (bits)	**RSA and Diffie-Hellman Key Size (bits)**	**Elliptic Curve Key Size (bits)**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes [1].

To use RSA or Diffie-Hellman to protect 128-bit AES keys one should use 3072-bit parameters: three times the size in use throughout the Internet today. The equivalent key size for elliptic curves is only 256 bits. One can see that as symmetric key sizes increase the required key sizes for RSA and Diffie-Hellman increase at a much faster rate than the required key sizes for elliptic curve cryptosystems. Hence, elliptic curve systems offer more security per bit increase in key size than either RSA or Diffie-Hellman public key systems.

Security is not the only attractive feature of elliptic curve cryptography. Elliptic curve cryptosystems also are more computationally efficient than the first generation public key systems, RSA and Diffie-Hellman. Although elliptic curve arithmetic is slightly more complex per bit than either RSA or DH arithmetic, the added strength per bit more than makes up for any extra compute time. The following table shows the ratio of DH computation versus EC computation for each of the key sizes listed in Table 1.

Security Level (bits)	**Ratio of DH Cost : EC Cost**
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Table 2: Relative Computation Costs of Diffie-Hellman and Elliptic Curves [2].

Closely related to the key size of different public key systems is the channel overhead required to perform key exchanges and digital signatures on a communications link. The key sizes for public key in Table 1 (above) is also roughly the number of bits that need to be transmitted each way over a communications channel for a key exchange. In channel-constrained environments, elliptic curves offer a much better solution than first generation public key systems like Diffie-Hellman.

In choosing an elliptic curve as the foundation of a public key system there are a variety of different choices. The National Institute of Standards and Technology (NIST) has standardized on a list of 15 elliptic curves of varying sizes. Ten of these curves are for what are known as binary fields and 5 are for prime fields. Those curves listed provide cryptography equivalent to symmetric encryption algorithms (e.g. AES, DES or SKIPJACK) with keys of length 80, 112, 128, 192, and 256 bits and beyond. For protecting both classified and unclassified National Security information, the National Security Agency has decided to move to elliptic curve based public key cryptography. Where appropriate, NSA plans to use the elliptic curves over finite fields with large prime modulo (256, 384, and 521 bits) published by NIST.

Attacks:

Attacks against the DH protocol come in a few flavors: –

-Denial of service Attacks: Here, the attacker will try to stop Alice and Bob from successfully carrying out the protocol. The attacker can accomplish this in many ways, for example by deleting the messages that Alice and Bob send to each other, or by overwhelming the parties with unnecessary computation or communication.

-Outsider Attacks: The attacker tries to disrupt the protocol (by for example adding, removing, replaying messages) so that he gets some interesting knowledge (i.e. information he could not have gotten by just looking at the public values).

-Insider Attacks: It is possible that one of the participants in a DH protocol creates a breakable protocol run on purpose in order to try to gain knowledge about the secret key of his peer. This is an important attack if one of the participants holds a static secret key that is used in many key agreement protocol runs. Note that malicious software could be very successful in mounting this attack.
The plausibility of these attacks depends on what assumptions we make about the adversary. For example, if the adversary can remove and replace any message from the public communication channel, the denial of service attack is impossible to prevent. Fortunately, it seems that complete breaks (outsider attacks in which the attacker obtains the shared secret key) and insider attacks can be prevented in many settings.

Conclusion:

The Diffie–Helman scheme is one of the exchanging key cryptosystem, no massages are involved in this scheme, in this report, and we try to benefit from this scheme by use the key (which exchange it) as a secret key. (That is, we know now the one of the advantages of the Diffie–Helman key exchange system) and we are using Elliptic curve cryptography for encryption and Decryption.

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman) now in use. As vendors look to upgrade their systems they should seriously consider the elliptic curve alternative for the computational and bandwidth advantages they offer at comparable security. Elliptic Curve Cryptography offers the highest strength-per-key-bit of any known public-key system of first generation techniques like RSA, Diffie-Hellman.ECC offers the same level of security with smaller key sizes, computational power is high. Integrated circuit space is limited for smart card, wireless devices. The ongoing development of standards is a very important position for the use of a cryptosystem. Standards help to ensure security and interoperability of different implementations of one cryptosystem.
There are several major organizations that develop standards like International Standards Organization (ISO), American National Standards Institute (ANSI), Institute of Electrical and Electronics Engineers (IEEE), Federal Information Processing Standards (FIPS).The most important for security in information technology are the in addition secure communication, Elliptic curve cryptography (ECC) enabling technology for numerous wireless sensor networks.

References:

[1] B.Schneier. Applied Cryptography. John Wiley and Sons, second edition, 2012.
[2] Cryptography and Elliptic Curves, koblitz, second edition, 2011.
[3] Julio Lopez and Ricardo Dahab, “An overview of elliptic curve cryptography”, May 2011.
[4] V. Miller, “Uses of elliptic curves in cryptography”, Advances in Cryptology - CRYPTO'85, LNCS 218, pp.417-426, 2011.
[5] Jeffrey L. Vagle, “A Gentle Introduction to Elliptic Curve Cryptography”, BBN Technologies, 2010

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 8, November 2015

[6] Mugino Saeki, “Elliptic curve cryptosystems”, M.Sc. thesis, School of Computer Science, McGill University, 2010.
[7] J. Borst, “Public key cryptosystems using elliptic curves”, Feb. 2010.
[8] Aleksandar Jurisic and Alfred Menezes, “Elliptic Curves and Cryptography”, Dr. Dobb's Journal, April 2010.
[9]Robert Milson, “Introduction to Public Key Cryptography, april 2009.
[10] Aleksandar Jurisic and Alfred J. Menezes, Elliptic Curves and Cryptography, 2008.