ON STUDY OF SOME ASYMMETRIC KEY CRYPTOGRAPHY ALGORITHMS

Mr. S. J. Chavhan

Assist. Professor,Dept. of Mathematics,Shri Vyankatesh college,D. raja (MH) India-443204

ABSTRACT

The cryptography uses different method by encrypting data at sender side and decrypted at receiver side .For securing the data there are two main types of cryptography algorithm, one is called symmetric and other is called asymmetric algorithms.

In this paper I discuss about some asymmetric key algorithms with its advantages and disadvantages. In particular RSA, DSA, Diffie-Hellman,Elgamal, Elliptic curve cryptography and XTR for asymmetric encryption algorithms.Asymmetric key algorithms use different keys for encryption and decryption. The keys arePrivate Key and Public Key. . Public key for encrypt (conversion form data or information to code) and private key for decrypt (reciprocal of encryption or reverse process of encryption) a message.Public key is very efficient for authentication as well as for encryption. The decryption key cannot be derived from the encryption key. Asymmetric key algorithms used for transmitting encryption keys or other data securely even when the parties have no opportunity to agree on a secret key in private.

Keywords: *Cryptography, Diffie-Hellman, RSA, ElGamal, ECC,DSA ,XTR.*

1.Introduction

Due to the extensive use and sharing of data in the Internet, it is necessary to protect data from hacking, noise, and interference. Cryptography attracted many researchers. It is used for protecting information during transmissions between users. It alters the content of transmitted data to unreadable form, once received by the receiver, it is converted back to its original form. Encrypting data results to an unreadable format called cipher-data. Reversion this cipherdata to original data called decryption process. Cryptography had a set of security goals to ensure the privacy of data. These goals are confidentiality, authentication, data Integrity, nonrepudiation and access control.Cryptography is widely used to secure data in cloud computing. It is classified into Symmetric (private-key) and Asymmetric (public key) keys encryption. Examples of Symmetric algorithms are DES, 3DES, AES, Blowfish and DSA (Digital Signature Algorithm), Elliptic Curve, DiffieHellman (key exchange) and RSA are examples of Asymmetric algorithms.

Asymmetric key algorithms are used for key distribution. Asymmetric key algorithms are also known as public key algorithms. Asymmetric key algorithms using two keys: A public key and a private key. Public key are used to encrypt the message and private keys are used to decrypt the message. Public key is known to public and private key is only known to user. So there is no need to distribute the keys before transmission. In this type of algorithms it is very difficult to derive one key from the other.

This paper work mainly focuses on brief description about various asymmetric key algorithms. Such as Diffie-Hellman, Rivest Shamir Adleman, ElGamal Encryption Algorith m, Elliptic Curve Cryptography and Digital Signature Algorithm.

2. Public-key encryption

A public-key encryption scheme (E, D) has two properties

1. (Completeness) Given any message m and key pair (K, k), the encryption function and decryption function are inverses: $E_K(D_k(m)) = D_k(E_K(m)) = m$.
2. (Semantic Security) For any pair of messages m and m', it is computationally hard to distinguish encryptions of m from encryptions of m', even given the public key of a principal.

3.Public Key Cryptography Algorithms

3.1 Rivest Shamir Adleman (RSA)

RSA is the most commonly used asymmetric algorithm. It can be used both for encryption and for digital signatures. RSA can be used for key exchange as well as digital signatures and the encryption of small blocks of data. Today, RSA is primary used to encrypt the session key used for secret key encryption or the message's hash value.

RSA mathematical hardness comes from the ease in calculating large numbers and the difficulty in finding the prime factors of those large numbers. Although employed with numbers using hundreds of digits, the math behind RSA is relatively straightforward.

Key algorithm:

1) Choose two prime numbers, p and q. From p and q you can calculate the modulus, n = pq
2) Select a third number, e, which is relatively prime to the product (p-1)(q-1). The number e is the public exponent.
3) Calculate an integer d from the quotient (ed-1) / [(p-1)(q-1)]. The integer number d is the private exponent.

RSA key lengths of 512 and 768 bits are considered to be pretty weak. For a reasonable level of security the key size should be greater than 1024 bits. 2048 bits key size should allow security for decades.

3.2 Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) has been suggested and standardized by the National Institute of Standards and Technology (NIST) of the U.S. the DSA defines the technique for generating and validating digital signatures. The DSA can be used by the recipient of a message to verify that the message has not been altered during transit as well as ascertain the originator identity. It is an electronic version of a written signature in that the digital signature can be used in proving to the recipient or a third party that the message was, in fact, signed by the originator. The digital signatures may also be generated for stored data and programs so that the integrity of the data and programs may be verified at any later time.

The DSA is used by a signatory to generate a digital signature on data and by a verifier to verify the authenticity of the signature. Every signatory has a public and private key. In the signature generation process the private key is used and the public key is used in the signature verification process.

Key algorithm:

1) Generate two large, distinct primes p, q .
2) Compute $n = pq$ and $\varphi(n) = (p-1)(q-1)$
3) Select a random number $1 < e < \varphi$ such that $\gcd(e, \varphi) = 1$
4) Compute the unique integer $1 < d < \varphi$ such that $ed \equiv 1 \bmod \varphi$
5) (d, n) is the private key
6) (e, n) is the public key p and q must be destroyed at the end of key generation.

Signature generation:

a. Compute $m^* = R(m)$ an integer in $[0, n-1]$
b. Compute $s = m^*d \bmod n$
c. A's signature for m is s

Signature verification:

1. Obtain A's authentic public key (e, n)
2. Compute $m^* = s^e \bmod n$

3. Verify that m* is in MR; if not reject the signature
4. Recover m = R-1(m*)

3.3 Diffie-Hellman (DH)

Diffie-Hellman is the first asymmetric encryption algorithm, invented in 1976 by Whitfield Diffie and Martin Hellman, using discrete logarithms in a finite field. It allows two users to exchange a secret key over an insecure medium without any prior secrets. Diffie-Hellman (DH) is a widely used key exchange algorithm.

The Diffie-Hellman protocol to be secure when an appropriate mathematical group is used. The generator element used in the exponentiations should have a large period. Usually, Diffie-Hellman is not implemented on hardware.

Key algorithm:

1) A and B publicly select a finite group G and an element α ϵ G.
2) A generates a random integer a, computes α^a in G, and transmits α^a to B through a public communications channel and vice versa.
3) A receives α^b and computes$(\alpha^b)^a$
4) B receives α^a and computes $(\alpha^a)^b$

3.4Elgamal Encryption Algorithm

ElGamal encryption system is an asymmetric key encryption algorithm for public key cryptography which is based on the Diffie-Hellman key exchange. It was described by Taher Elgamal in 1984. ElGamal is the predecessor of Digital signature algorithm. ElGamal encryption consists of three components: they are key generator, encryption algorithm, and the decryption algorithm .

Key Algorithm:

1) A finite cyclic group G of order n and generator α ϵ G are chosen. Each user picks a random integer $l\epsilon$ {0, 1, . . . , n − 1} (private key), and makes public α^l (public key). We suppose that messages are elements of G and that user A wishes to send a message m to user B.
2) A generates a random integer k ϵ {0, 1,. . . n − 1} and computes α^k.
3) A looks up B's public key α^l and computes $(\alpha^l)^k$. then mα^{lk}.
4) A sends to B the pair of group elements (αk,mα^{lk}).
5) B computes (mα^{lk}) $((\alpha^l)^k)^{-1}$ = m$\alpha^{lk}(\alpha^{lk})^{-1}$= m and recovers the message.

ElGamal encryption is probabilistic, i.e. a single plaintext can be encrypted to many possible cipher texts, with the consequence that a general ElGamal encryption produces a 2:1 expansion in size from plaintext to cipher text. For encryption ElGamal requires two exponentiations. These exponentiations are independent of the message and can be computed ahead of time if need be. Decryption requires only one exponentiation.

3.5 Elliptic Curve Cryptography (ECC)

ECC was introduced by Victor Miller and Neal Kolbitz as an alternative to established public key systems such as RSA . In 1985, they proposed a public key cryptosystems analogue of ElGamal encryption schema witch used Elliptic Curve Discrete Logarithm Problem (ECDLP) . Elliptic curve cryptosystems (ECCs) include key distribution, encryption algorithms. The key distribution algorithm is used to share a secret key and the encryption algorithm enables confidential communication. ECC is based on the addition of rational points on a chosen elliptic curve.

One of the main benefits in comparison with non-ECC cryptography is the same level of security provided by keys of smaller size. Elliptic curves are applicable for the encryption, digital signatures, pseudo-random generators and other tasks. It is also used in several integer factorization algorithms that have applications in cryptography.

XTR

XTR is an algorithm for asymmetric encryption (public-key encryption). XTR is a novel method that makes use of traces to represent and calculate powers of elements of a subgroup of a finite field. It is based on the primitive underlying the very first public key cryptosystem, the Diffie-Hellman key agreement protocol. From a security point of view, XTR security relies on the difficulty of solving discrete logarithm related problems in the multiplicative group of a finite field. Some advantages of XTR are its fast key generation (much faster than RSA), small key sizes (much smaller than RSA, comparable with ECC for current security settings), and speed (overall comparable with ECC for current security settings).

4.Advantages and Disadvantages

The advantages and disadvantages of various asymmetric key cryptography algorithms are discussed in the following table.

Table 1. Advantages and disadvantages of various asymmetric key cryptography algorithms.

Sr.No.	Algorithms	Advantages	Disadvantages
01	Diffie Hellman (DH)	▪ The sender and receiver have no prior knowledge of each other. ▪ Communication can take place through an insecure channel. ▪ Sharing of secret key is safe.	▪ Can not be used for asymmetric key exchanging. ▪ Cannot be used for digital signature. ▪ Vulnerable to main in the middle attacks science it does not authenticate either party involved in exchange.
02	Rivest Shamir Adleman (RSA)	▪ RSA algorithm is safe and secure for its users through the use of complex mathematics. ▪ RSA algorithm is hard to crack since it involves factorization of prime numbers which are difficult to factorize. ▪ Moreover, RSA algorithm uses the public key to encrypt data and the key is known to everyone, therefore, it is easy to share the public key.	▪ RSA algorithm can be very slow in cases where large data needs to be encrypted by the same computer. ▪ It requires a third party to verify the reliability of public keys. ▪ Data transferred through RSA algorithm could be compromised through middlemen who might temper with the public key system
03	ElGamal	▪ Based on discrete logarithm problems. ▪ Security as that of DH system. ▪ DL problem needs less key bits than RSA for the same security. ▪ Asymmetric workload: good for some applications.	▪ The cryptosystem needs more bits than the plaintext(double). ▪ A new random is needed for every encrypted message. ▪ Asymmetric workload: bad for some applications.
04	Elliptic Curve Cryptography (ECC)	▪ Greater exibility, smaller key sizes with the same security. ▪ ECC does not require prime numbers and exponential processing for encryption. ▪ ECC offers considerable bandwidth savings when being used to transform short messages.	▪ Hyper elliptic cryptosystems over even smaller key sizes. ▪ ECC is Mathematically more difficult to explain/justify to the client. ▪ Confidence level in ECC is not as high as RSA.

05	Digital Signature Algorithm (DSA)	▪ Speed: Businesses no longer have to wait for paper documents to be sent by courier. Contracts are easily written, completed, and signed by all concerned parties in a little amount of time no matter how far the parties are geographically. ▪ Costs: Using postal or courier services for paper documents is much more expensive compared to using digital signatures on electronic documents. ▪ Security: The use of digital signatures and electronic documents reduces risks of documents being intercepted, read, destroyed, or altered while in transit. ▪ Authenticity: An electronic document signed with a digital signature can stand up in court just as well as any other signed paper document. ▪ Tracking: A digitally signed document can easily be tracked and located in a short amount of time. ▪ Non-Repudiation: Signing an electronic document digitally identifies you as the signatory and that cannot be later denied. ▪ Imposter prevention: No one else can forge your digital signature or submit an electronic document falsely claiming it was signed by you. ▪ Time-Stamp: By time-stamping your digital signatures, you will clearly know when the document was signed.	▪ Expiry: Digital signatures, like all technological products, are highly dependent on the technology it is based on. In this era of fast technological advancements, many of these tech products have a short shelf life. ▪ Certificates: In order to effectively use digital signatures, both senders and recipients may have to buy digital certificates at a cost from trusted certification authorities. ▪ Software: To work with digital certificates, senders and recipients have to buy verification software at a cost. ▪ Law: In some states and countries, laws regarding cyber and technology-based issues are weak or even non-existent. Trading in such jurisdictions becomes very risky for those who use digitally signed electronic documents. ▪ Compatibility: There are many different digital signature standards and most of them are incompatible with each other and this complicates the sharing of digitally signed documents.
06	XTR	▪ XTR is secure, efficient, compact, easy to implement, with trivial parameter generation.	▪ Do we really trust GF(p^6)? ▪ Multiplication of Tr(g^m) $and\ Tr(g^n)$ is non-trivial. ▪ p^6 grows as fast as RSA moduli i.e. fast. ▪ q grows as fast as ECC subgroups i.e. Slow.

5. Conclusion

In this paper we study that from the security point of view the Asymmetric Cryptography technique is more secure than the symmetric cryptography. Because asymmetric cryptography is difficult to break the system, that

uses two different keys.we try to analyse, ensure the confidentiality and authentication in information during sharing through cryptography of asymmetric key. As we discussed that confidentiality mean protection of message from observer and authentication mean that receiver needs assurance as the identity of sender. In most practical implementations asymmetric key cryptography is used to secure and distribute session keys. We the basic asymmetric key algorithms, key concepts, security are presented .

References

[1]. W. Stallings, Cryptography and Network Security Principles and Practices Fourth Edition, Pearson Education, Prentice Hall, 2009.

[2]. M. Preetha, M. Nithya, June 2013, A Study And Performance Analysis Of RSA Algorithm, IJCSMC, Vol. 2, Issue. 6, pg.126 – 139.

[3]. Alese, B. K.Philemon E.D., Falaki, S. O., September 2012 , Comparative Analysis of Public-Key Encryption Schemes, International Journal of Engineering and Technology, Volume 2, No. 9.

[4]. Dr.Prerna Mahajan, Abhishek Sachdeva, 2013 , A study of Encryption algorithms AES, DES and RSA for security, Global Journal of Computer Science and Technology, Volume 13 Issue 15 Version 1.0.

[5]. Certicom Corp., 2004, An elliptic curve cryptography (ECC) primer, White paper, Certicom.

[6]. R. L. RIVEST, A. SHAMIR, AND L. ADLEMAN, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 21 (1978), pp. 120–126.

[7] T. E. GAMAL, A public key cryptosystem and a signature scheme based on discrete logarithms,

[8]. S. Vijaykumar and S. Saravanakumar, 2011, Future Robotics Database Management System along with Cloud TPS, Intl. Journal on CloudComputing: Services and Architecture (IJCCSA), pp. 103–114.

[9]. Rashmi Singh, Shiv Kumar, December 2012, ElGamal Algorithm in Cryptography, International Journal of Scientific & Engineering Research, Volume 3, Issue 12.

[10] V. S. Miller, 1986, Use of Elliptic Curves in Cryptography, Advances in Cryptology Crypto'85, LNCS.218, New York, Springer-Verlag, pp. 417-426.