Amalna Jose* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.1, December - January 2016, 2704 – 2709.

# A Literature Study on Parallel Key Cryptographic Algorithm

**AMALNA JOSE**
M.Tech-Student
Computer Science and Engineering,
R V College of Engineering,
Bangalore, Karnataka, India.

*Abstract:* **In the field of computer security there are a large number of papers discussing on the topic of cryptography. Cryptography is an art of sending data to the intended recipient by preserving the integrity, confidentiality and authenticity of the data. Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with converting plain-text (ordinary text, also referred as clear-text) into cipher-text (by a process called encryption), then back again (by a process known as decryption) to plain-text that is the original message. The main objectives of cryptography are Confidentiality (the message cannot by understood by anyone other than the intended recipient), Integrity (the message cannot be altered during its storage or transmission.), Non- repudiation (the creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information), Authentication (the sender and receiver can confirm each other's identity and the origin/destination of the information).**

*Key words:* **Cryptography, Microdots, Merging, Encryption, Decryption**

## I. INTRODUCTION

In cryptography, there are two types of mechanisms based on the usage of key: symmetric key, and asymmetric-key. Symmetric key is an encryption system in which both the sender and the receiver of a message share a single, a common key that is used to encrypt and decrypt the message. Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Asymmetric key is a cryptographic system that uses two keys a public key known to everyone and a private or secret key known only to the recipient of the message. For example when John wants to send a secure message to Jane, he uses Jane's public key to encrypt the message. Jane then uses her private key to decrypt it. An important element of the public key system is that the public and private keys are related in such a way that if a public key is used to encrypt messages then only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key. However asymmetric key has a big disadvantage of the complexity involved in the encryption and decryption process. Hence the algorithms are mainly used to encrypt or decrypt shorter messages. In the study of parallel encryptions, it can perform applications e.g. message authentication (signcryption). For instance, some of them are: the solutions of solving key agreements or key exchanges problem, and the cryptographic algorithm for Color Images were proposed in[1]. In modern information block cipher, the decryption algorithm must be the inverse of encryption algorithm, and both algorithms have to use the corresponding key to each other (for asymmetric-key) so that receiver can recover the plaintext sent by sender. There is a mode of operation, called "Cipher Block Chaining (CBC)", which each plaintext block is exclusive-ored with the previous ciphertext block. This method can prevent an algorithm from replay attack which is the main method in attacking of traditional Electronic Code Book (ECB).

### *Parallel Key Cryptographic Algorithm*

In cryptography, asymmetric-key is the framework with high flexibility which can be applied to most cryptographic systems in use today. To reduce the encryption and decryption time which is the main drawback of asymmetric-key cryptography, a new mechanism called "Parallel key Cryptographic Algorithm (PCA)" is proposed. The algorithm accelerates the cryptographic system in encryption and decryption process and strengthens the system against Brute force attack. Experimental results have shown that, the proposed algorithm can perform faster in encryption and decryption of messages when compared to a famous asymmetric-key cryptographic algorithm; RSA (Rivest, Shamir and Adleman). In the theoretical analysis, it is shown that PCA can provide security against Brute force attack better than other algorithms. Furthermore, PCA can be applied to parallel cryptographic mode such as Cipher Block Chaining (CBC) and Interleaved Cipher Block Chaining (ICBC) for higher efficiency.

  

## II. RESEARCH WORK RELATED TO CRYPTOGRAPHY

There are many active threads in cryptography research. Many of them are where cryptography meets a particular branch of mathematics (number theory, algebraic geometry, complexity theory, combinatorics, graph theory, and information theory). The empirical end of the business is concerned with designing primitives for encryption, signature, and composite operations, and that perform reasonably well on available platforms. The two meet in the study of subjects ranging from linear and differential cryptanalysis to attacks on public key protocols. Research is more driven by the existing body of knowledge than by applications, though there are exceptions: copyright protection concerns have been a stimulus, and so has the recent competition to find an Advanced Encryption Standard. Current research at the theoretical end of cryptology is found at the FOCS, STOC, Crypto, Eurocrypt, and Asiacrypt conferences.

The main steps involved in the algorithm are as follows:-

1) Key generation: a couple of key pairs are generated where the key length for first key is more than that of the second key.
2) Encryption: Process where the plain text is converted to cipher-text using the parallel public keys generated.
3) Decryption: Process where the cipher-text is converted back to plain text using the parallel private keys generated.

## III. RESEARCH PAPERS RELATED TO CRYPTOGRAPHY

Umpteen number of papers were reviewed related to cryptography and parallel key algorithms some of the papers include the following :- RSA cryptosystems was conceptualized designed and developed by R. Rivest, A. Shamir, and L. Adleman. To provide Privacy of the data and providing the authority to the owner of digital data are the main factors provided by RSA. DH/DSA A new algorithm based upon the public key cryptography was developed called public key Digital Signature Algorithm was invented by Diffie and Hellman which is purely on the application of discrete log problem over a prime field. In the early 90s the NIST of united states came up with a signature standard which was based upon this algorithm. Junfeng Fan, Kazuo Sakiyama and Ingrid Verbauwhede (2009) In this paper they have implemented Elliptic Curve Cryptography (ECC) on an multi core embedded system, and the methods for scheduling the task at various kinds of levels has been studied in detail. A method for scheduling instructions that makes use of the cores to carry out functions in parallel a single modular operation has been proposed. STATE OF THE ART PARALLEL APPROACHES FOR RSA PUBLIC KEY BASED CRYPTOSYSTEM Sapna Saxena and Bhanu Kapoor February 2015 In this paper a survey of various parallel implementations of RSA algorithm involving variety of hardware and software implementations are proposed. Lejla Batina, Alireza Hodjat, David Hwang, Kazuo Sakiyama and Ingrid Verbauwhed (2006) This paper discusses different architectures that will suit the employment of cryptographical services to enhance the security and protection of the system taking in to consideration the lowest cost possible.

***Asymmetric Key Cryptography***

Public-key cryptography or asymmetric key cryptography refers to a set of cryptographic algorithms that are based on mathematical problems that currently admit no efficient solution -- particularly those inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate a public and private key-pair and to use it for encryption and decryption. The strength lies in the "impossibility" (computational impracticality) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security. Security depends only on keeping the private key private. Public key algorithms, unlike symmetric key algorithms, do not require a secure channel for the initial exchange of one (or more) secret keys between the parties. Because of the computational complexity of asymmetric encryption, it is typically only used for short messages, typically the transfer of a symmetric encryption key. This symmetric key is then used to encrypt the rest of the potentially long & heavy conversation. The symmetric encryption/decryption is based on simpler algorithms and is much faster. Two of the best-known uses of public-key cryptography are: Public-key encryption, in which a message is encrypted with a recipient's public key. The message cannot be decrypted by anyone who does not possess the matching private key, who is thus presumed to be the owner of that key and the person associated with the public key. This is used in an attempt to ensure confidentiality. Second one is the Digital signatures, in which a message is signed with the sender's private key and can be verified by anyone who has access to the sender's public key. This verification proves that the sender had access to the private key, and therefore is likely to be the person associated with the public key. This also ensures that the message has not been tampered with, as any manipulation of the message will result in changes to the encoded message

  

digest, which otherwise remains unchanged between the sender and receiver.

***RSA (Rivest-Shamir-Adleman)***

RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, only the intended recipient can decrypt the message using the secret private key. RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total -- factoring -- is considered infeasible due to the time it would take even using today's super computers. The public and the private key-generation algorithm is the most complex part of RSA cryptography. Two large prime numbers, p and q, are generated using the Rabin-Miller primality test algorithm. A modulus n is calculated by multiplying p and q. This number is used by both the public and private keys and provides the link between them. Its length, usually expressed in bits, is called the key length. The public key consists of the modulus n, and a public exponent, e, which is normally set at 65537, as it's a prime number that is not too large. The e figure doesn't have to be a secretly selected prime number as the public key is shared with everyone. The private key consists of the modulus n and the private exponent d, which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the totient of n.

Available Open Source Tools for Cryptography

***1. OpenSSH***

OpenSSH, also known as OpenBSD Secure Shell, is a suite of security-related network-level utilities based on the SSH protocol, which help to secure network communications via the encryption of network traffic over multiple authentication methods and by providing secure tunneling capabilities.[12]

***2. GNU Privacy Guard (GnuPG or GPG)***

GNU Privacy Guard (GnuPG or GPG) is a free software replacement for Symantec's PGP cryptographic software suite. GnuPG is compliant with RFC 4880, which is the IETF standards track specification of OpenPGP. Modern versions of PGP and Veridis' Filecrypt are interoperable with GnuPG and other OpenPGP-compliant systems. GnuPG is part of the GNU project, and has received major funding from the German government.[13]

***3. OpenSSL***

In computer networking, **OpenSSL** is a software library to be used in applications that need to secure communications against eavesdropping or need to ascertain the identity of the party at the other end. It has found wide use in internet web servers, serving a majority of all web sites.

OpenSSL contains an open-source implementation of the SSL and TLS protocols. The core library, written in the C programming language, implements basic cryptographic functions and provides various utility functions. Wrappers allowing the use of the OpenSSL library in a variety of computer languages are available.

***4. Signal***

Signal is a free and open-source encrypted voice calling and instant messaging application for iOS and Android. It uses advancedend-to-end encryption protocols to secure all communications to other Signal users. Signal can be used to send and receive encrypted instant messages, group messages, attachments and media messages. Users can independently verify the identity of their messaging correspondents by comparing key fingerprints out-of-band. During calls, users can check the integrity of the data channel by checking if two words match on both ends of the call.

***5. Seahorse***

**Seahorse** is a GNOME front-end application for managing PGP and SSH keys. Seahorse integrates
with Nautilus, gedit andEvolution for
encryption, decryption and other operations. It has HKP and LDAP key server support. The program is based onGNU Privacy Guard (GPG) and is released as free software under the GNU General Public License (GPL).

***6. GNU TLS***

GnuTLS (the GNU Transport Layer Security Library) is a free software implementation of the TLS, SSL and DTLS protocols. It offers an application programming interface (API) for applications to enable secure communication over the network transport layer, as well as interfaces to

  

access X.509, PKCS #12, OpenPGP and other structures.

***7. Open VPN***

OpenVPN is an open-source software application that implements virtual private network (VPN) techniques for creating secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities. It uses a custom security protocol[14] that utilizes SSL/TLS for key exchange. It is capable of traversing network address translators (NATs) and firewalls. It was written by James Yonan and is published under the GNU General Public License (GPL)[15].

***8. AxCrypt***

AxCrypt is a software for casewise encryption and decryption of single files, which the AES - algorithm with a key length of 128 bits used. AxCrypt generated an archive, which in addition to the encrypted data file contains additional metadata and then deletes the original file when encrypting a file. To open the archive, a double use. AxCrypt then decrypts the file containing these writes to the file system and then calls therefore responsible for the file extension on application.

***9. EncFS***

EncFS is a Free (LGPL) FUSE-based cryptographic filesystem. It transparently encrypts files, using an arbitrary directory as storage for the encrypted files.

Two directories are involved in mounting an EncFS filesystem: the source directory, and the mountpoint. Each file in the mountpoint has a specific file in the source directory that corresponds to it. The file in the mountpoint provides the unencrypted view of the one in the source directory. Filenames are encrypted in the source directory.

***10. CrossCrypt***

It is an open-source on-the-fly encryption program for the Microsoft Windows XP/2000 operating systems. CrossCrypt allows a user to make virtual drives which encrypt any files stored on them, making the encryption process completely seamless to the user. CrossCrypt is based on FileDisk, virtual disk driver for Windows NT/2000/XP that uses one or more files to emulate physical disks, adding encrypted volumes functionality.

***Licensed Tools for Cryptography***

***1. Folder Lock***

Folder Lock is a fast data encryption and password protection software for Windows. It can simultaneously encrypt, lock and password protect your files, folders, drives, USB drives and even CD/DVD-RW.Folder Lock creates encrypted storages called 'Lockers'. You can keep as many of your private files & folders in your Locker and password protect it with a single click. You can transfer, secure and backup these Lockers. Lockers are portable, you can keep them in USB Flash Drives, CD/DVD (R-RW), & notebooks or transfer them via email or upload. Lockers are even undeletable on the computer where Folder Lock is installed.

***2. File Encryption XP***

File Encryption XP encrypts files and folders using a strong Blowfish algorithm with 384-bit key. Protected files that can be decrypted without File Encryption XP is a built in program feature. You can create a self-extracting encrypted file and simply send it by mail or give it to someone on a floppy disk. If the recipient knows the password, he or she can execute this file to get the original document.

***3. Crypto Forge***

CryptoForge is easy-to-use file encryption software for personal and professional data security. It allows you to protect the privacy of sensitive files, folders, or email messages, by encrypting them with up to four strong encryption algorithms. Once the information has been encrypted, it can be stored on insecure media or transmitted on an insecure network -like the Internet- and still remain secret. Later, the information can be decrypted into its original form. CryptoForge integrates the strongest encryption available today into the Windows environment.

***4. Safe House***

SafeHouse provides total privacy and protection for your sensitive files and folders using passwords and strong encryption. SafeHouse features military-strength encryption which is completely transparent to the way you work and compatible with all Windows applications.

***5. Ketu File***

KetuFile is a file encryption program.You can use KetuFile to encrypt any file on a

  

Windows(tm) PC, such as word processor files, spreadsheets, images, etc.Once a file is encrypted it can only be read by someone who has the secret Key.You can use KetuFile to encrypt files up to 2 gigabytes in length.KetuFile uses Advanced Encryption Standard (AES), which is the official encryption system for the U.S. government.KetuFile will encrypt with up to 512 bits Key length. We believe this to be the strongest commercially available encryption in the world.

***6. Sensi Guard***

SensiGuard provides robust file and folder encryption, which means that you will never have to worry about your private information leaking out. Your credit card data, personal emails, tax returns, photos and other private files will remain under lock and key.

***7. SafeBit***

SafeBit is a disk encryption software and the perfect solution for protecting your data against potential unauthorized access and information leaks. It features on-the-fly disk encryption, by creating virtual disk drives, where you can hide files & folders, keep them encrypted all the time, but still work with these files just like you work with normal files.

***8. Dekart Keeper***

Encrypt files and directories with a password or a keyDesigned to be extremely easy to use, it never gets in your way while you do your job. It memorizes all the strong passwords for you, so you don't have to type them. Keeper will remember the passwords that you use for different people or groups of people.All the frequently used features are at your fingertips – wipe files, encrypt and email your data with a few clicks.

***9. Secure IT***

SecureIT is a smartly designed encryption utility with large, iconic buttons and a Windows-style Explorer interface. Its main features are encryption, decryption, compression and deleted-file shredding. There are two encryption algorithms to choose from: 256-bit AES or 448-bit Blowfish. So whichever side you come down on in the encryption algorithm religious debate, you can use one or the other. For example, if you prefer to use the same method that the U.S. federal government uses, then you have AES. But if you believe the rumors about AES containing some sort of NSA backdoor, then you have Blowfish. SecureIT hides the complexities of encryption so you can benefit from the functionality without a graduate degree in mathematics.

***10. Cryptrainer PE***

Cryptainer PE protects your data by creating multiple encrypted vaults for all your files and folders. Secure your PC and ensure total privacy with Cryptainer PE's powerful 448 bit strong encryption without changing the way you work. Offers choice of Blowfish, as well as, AES encryption algorithms. You can create a virtual drive of upto 25000 MB, that can be loaded and unloaded as required, with a password. You can drag and drop any kind of data that you wish to protect, into the cryptainer drive. Cryptainer PE's secure email module allows you to generate encrypted files that can be sent as email attachments.

## IV. ACKNOWLEDGEMENTS

The author thanks Dr.S.Sridhar, Professor and Dean, Cognitive & Central Computing, R.V.College of Engineering, Bangalore, India for communicating this article to this Journal for publication. The author also thanks Dr.G.Shobha, Professor and Head, Department of CSE, R.V.College of Engineering, Bangalore, India for the support given and Dr.N.K.Srinath, Professor and Dean(Students' affairs) and Prof.S.Sandhya , CSE department for technical advice and guidance.

## V. REFERENCES

**[1].** Archana B.Kadga, Implementing Asymmetric-Key Cryptography using Parallel-Key Cryptographic Algorithm (PCA), International Journal of Computer Science and Mobile Applications,Vol.1 Issue. 6, December- 2013,

**[2].** Behrouz A. Forouzan, Cryptography and Network Security, McGraw- Hill International Edition, 2008

**[3].** R. M. Dansereau, S. Jin, R. A. Goubran, "Reducing Packet Loss in CBC Secured VoIP using Interleaved Encryption", Proceeding of IEEE CCECE/CCGEI, Ottawa, May, 1-4244-0038-4 , 2006

**[4].** Ying Wang, Chunyan Han and Yuanyi Liu, "A Parallel Encryption Algorithm for Color Images Based on Lorenz Chaotic Sequences", IEEE transaction on information Theory, 1-4244-0332-4/06, pp.9744- 9747,2006

**[5].** Yun Chen, Xin Chen, YiMu,"A Parallel Key Generation Algorithm for Efficient Diffie-Hellman Key Agreement", IEEE Transactions on information Theory, 1-4244-0605-6/06, pp.1393– 1395, 2006

  

**[6].** David Pointcheval, Jacques Stern, "Security Proofs for Signature Schemes", E'cole Normale Supe'rieure 2005.

**[7].** Atul Kahate, CRYPTOGRAHY and NETWORK SECURITY, Tata McGraw-Hill Publishing Company Limited, 2003

**[8].** Praveen Dongara and T. N. Vijaykumar, "Accelerating Private-Key Cryptography via Multithreading on Symmetric Multiprocessors", Proceeding of IEEE Int'l Symp. Performance Analysis of Systems and Software (ISPASS 03), IEEE Press, 2003.

**[9].** Schneier, Bruce, Applied Cryptography Second Edition: protocols, algorithms, and source code in C, John 1996.

**[10].** Dan Boneh and Ramarathnam Venkatesan, Breaking RSA may be easier t2005han factoring, Advances in Cryptology EUROCRYPT '98 (Kaisa Nyberg, ed.), LNCS, no. 1403, IACR, Springer, May 1998, pp. 59–71.

**[11].** Benoˆıt Libert, Jean-Jacques Quisquater, and Moti Yung, "Parallel Key- Insulated Public Key Encryption Without Random Oracles", UCL, Micorelectronics Laboratory, Crypto Group (Belgium) and RSA Labs and Columbia University (USA)

**[12].** Venkatachalam, Girish (April 2007). "The OpenSSH Protocol under the Hood". Linux Journal (156): 74–77Accessed via the Discovery Database at LSU

**[13].** "Bundesregierung fördert Open Source" (in German). Heise Online. 1999-11-15. Retrieved July 24, 2013.

**[14].** "OpenVPN Security Overview". Retrieved 28 September2011.

**[15].** LinuxSecurity.com - OpenVPN: An Introduction and Interview with Founder, James Yonan

**[16].** Menezes, A. J., Vanstone, S. A., & Oorschot, P. C. V. 1996. Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA.

**[17].** https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/Public-key-crypto-1.svg/288px-Public-key-crypto-1.svg.png

**[18].** https://upload.wikimedia.org/wikipedia/commons/f/f9/Public_key_encryption.svg

**[19].** Diffie, W. & Hellman, M. Nov 1976. New directions in cryptography. Information Theory, IEEE Transactions on, 22(6), 644–654.

**[20].** R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and publickey cryptosystems. Communications of the ACM, 21(2):120{126, 1978.

**[21].** Bajard, J., and Laurent Imbert. "A full RNS implementation of RSA." Computers, IEEE Transactions on 53, no. 6 (2004): 769-774.

**[22].** Rawat, Anshuman, and Shabsi Walfish. "A Parallel Signcryption Standard using RSA with PSEP." (2003).

**[23].** Ciet, Mathieu, Michael Neve, Eric Peeters, and J-J. Quisquater. "Parallel FPGA implementation of RSA with residue number systems-can side-channel threats be avoided?." In Circuits and Systems, 2003 IEEE 46th Midwest Symposium on, vol. 2, pp. 806-810. IEEE, 2003

**[24].** Tang, S. H., K. S. Tsui, and Philip Heng Wai Leong. "Modular exponentiation using parallel multipliers." In Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on, pp. 52-59. IEEE, 2003.

**[25].** Wu, Chia-Long, Der-Chyuan Lou, Jui-Chang Lai, and Te-Jen Chang. "Fast parallel exponentiation algorithm for RSA public-key cryptosystem." Informatica 17, no. 3 (2006): 445-462

**[26].** Liu, Qiang, Fangzhen Ma, Dong Tong, and Xu Cheng. "A regular parallel RSA processor." In Circuits and Systems, 2004. MWSCAS'04. The 2004 47th Midwest Symposium on, vol. 3, pp. iii- 467. IEEE, 2004.

**[27].** Chang, Weng-Long, Minyi Guo, and Michael Shan-Hui Ho. "Fast parallel molecular algorithms for DNA-based computation: factoring integers." NanoBioscience, IEEE Transactions on 4, no. 2 (2005): 149-163.

  