بسم الله الرحمن الرحيم

Islamic University – Gaza

Deanery of Graduate Studies

Faculty of Information Technology

الجامعة الإسلامية – غزة

عمادة الدراسات العليا

كلية تكنولوجيا المعلومات

Secure key Agreement for LAN Based on Multi-level Encryption Over GSM

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology

By

Mustafa Abu Saqer

120110458

Dr. Ashraf Alattar

Supervisor

(1437 H) February, 2016

إقــــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Secure key Agreement for LAN Based on Multi-level Encryption Over GSM

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name:	اسم الطالب/ة: مصطفى حسين أبوصقر
Signature:	التوقيع:
Date:	التاريخ:28 /5/ 2016

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا **هاتف داخلي:** 1150

الرقم ج س غ/35/ .... Ref
2016/03/09
التاريخ .... Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ مصطفى حسين حمد أبوصقر لنيل درجة الماجستير في **كلية *تكنولوجيا المعلومات*** **برنامج تكنولوجيا المعلومات** وموضوعها:

آلية توافق على مفتاح التشفير للشبكات المحلية (LAN) عبر شبكات الهواتف الخلوية (GSM) باستخدام التشفير متعدد المراحل

Secur Key Agreement for LAN Based on Multi-Level Encryption over GSM

وبعد المناقشة التي تمت اليوم الأربعاء 29 جمادى الأولى 1437هـ، الموافق 2016/03/09م الساعة الثامنة والنصف صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. أشرف محمد العطار مشرفاً و رئيساً ....

د. توفيق سليمان برهوم مناقشاً داخلياً ....

د. حسن نجيب قنوع مناقشاً خارجياً ....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية *تكنولوجيا المعلومات*/ برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

نائب الرئيس لشئون البحث العلمي والدراسات العليا

شئون البحث العلمي والدراسات العليا
Research & Graduate Affairs

أ.د. عبدالرؤوف علي المناعمة

Dedication

To my father

To my family

To my teachers

To my friends

To my colleagues

To Palestine

Acknowledgment

I thank Allah for giving me the strength and ability to complete this study despite all the difficult circumstances. I would like to express my sincere gratitude to my advisor Dr. Ashraf Alattar, without his help, guidance, and continuous follow-up; either Dr. Tawfiq Barhoom, or Dr. Hassan Qanooe this research would never have been.

Also I would like to extend my thanks to the academic staff of the Faculty of Information Technology who helped me during my Master’s study and taught me different courses. I cannot forget to express my thanks to the networks administrator, system securing administrator, and whole development staff in Government Computer Center and Ministry of Telecom and Information technology. I would like to thank my colleagues and classmates for making my study a great experience, useful, enjoyable, and full of warm atmosphere. Last but not least, I am greatly indebted to my family for their support during my course studies and during my thesis work.

Abstract

Encryption key transferring over a communication network is unsafe method, in the way it breaches the safety criteria once the key is discovered, this breaching conflicts with CIA triangle, specifically with confidentiality concept, where the data is already known by unauthorized party. Available encryption techniques might be one of the solutions which are offered to overcome the confidentiality concept of CIA triangle. However, the key transfer should be safe, and deny unauthorized parties to get the transferred key, except the parties that are authorized in the secured communication process. Securing the key data implemented with VPN, third part or key agreement before establish connection.

Our research is based on transferring the key over a network which will not be used for the communication process. The main idea is to generate two secret keys, the first is used as a secret key to encrypt the second key in multi-level encryption process, the encrypted second key will be sent over GSM network. The second key will be decrypted then used as a secret key for the text transferring over LAN. The first key will be generated using Diffie-Helman algorithm. This key will be used as a secret key for encrypt the second key over multi-level encryption process, which is implemented by passing the first key as secret key and encrypting the second key in a predefined order of algorithms where is the start index related to the connection request time stamp, the resulted key will be sent over GSM.

We have selected five encryption algorithms in the same order in the system. The selection of the first algorithm to start the multi-level encryption process is based on the connection time stamp. We have developed a pilot simulation for the solution, and got many improvements for the new solution which is summarized in: 1. reducing the required time of connection, instead of waiting for IO for knocking time and SMS sending receiving time we minimize the knocking for one request and the SMS for one time waiting .2. Financial connection establishment cost by decreasing 50%. Increasing the complexity of guessing the key and the algorithm to decrypt the sniffed data. In addition, controlling the key length by the algorithm settings.

Keywords Cryptography, Cryptanalysis, Key Generation, Key Distribution, Key Length.

عنوان البحث

توزيع مفتاح الأمان للشبكة المحلية بالاعتماد على تعدد طبقات التشفير على شبكة الهاتف النقال

الملخص

ان نقل مفتاح التشفير على الشبكات التي يتم التواصل خلالها يعتبر غير آمن نظرا للخطورة التي تنعكس على معايير الأمان في حال كشف المفتاح والتي ترتبط بالخصوصية أو تكامل البيانات وهي أحد أهم نقاط مثلث الأمان (مدى الوثوق من البيانات المنقولة، ومدى صحة نقلها وأخيرا أنها متاحة طوال الوقت). استخدام تقنيات التشفير المتاحة ربما يكون أحد الحلول لمشكلة الوثوق في البيانات المنقولة وكذلك تكامليتها، ولكن نقل المفتاح المتفق عليه بين طرفي التشفير هو المشكلة الرئيسة حيث انه يجب نقل المفتاح دون الوصول إليه من طرف آخر سوى طرفي التشفير.

يعتمد البحث الذي قمنا عليه باستخدام شبكة أخرى لنقل المفتاح بشكل مشفر في حين تم التنصت عليه لا يمكن الوصول لقيمته، وهنا نستخدم طريقتين للتشفير: الأولى لتشفير المفتاح المنقول والثانية لتشفير البيانات باستخدام المفتاح المنقول.

لنقل المفتاح الأول بشكل آمن نستخدم خوارزمية Diffie-Helman وبناء عليه يتم تشفير المفتاح بعدة طبقات تشفير وإرساله عبر GSM شبكة الهاتف النقال. ومن ثم يستخدم هذا المفتاح بعد فك تشفيره -بعكس ترتيب الطبقات بناء على تزامن وقتي بين الطرفين- لتشفير البيانات باستخدام خوارزمية عشوائية من طبقات التشفير ويتم فك تشفيرها بنفس الخوارزمية اعتمادا على التزامن بين طرفي الاتصال.

قمنا بإدراج ترتيب معين لخمسة خوارزميات تشفير يتم انتقاء نقطة البداية بناء على التزامن بين طرفي الاتصال ومن ثم يقوم برنامج المحاكاة بتمرير المفتاح على كافة الخوارزميات ومن ثم إرساله عبر الجوال للطرف الذي طلب الاتصال. يقوم المستقبل بوصل الجوال بالجهاز وقراءة المفتاح وعكس عملية التشفير بناء على نفس المزامنة والوصول للمفتاح المرسل. يبدأ طرفي الاتصال بإرسال الرسائل وبناء على اللحظة الزمنية للإرسال تستخدم أحد تلك الخوارزميات للتشفير ويرسل النص مشفر إلى الطرف الثاني ويتم عكس التشفير والوصول للمحتوى. بعد تنفيذ المحاكاة وصلنا الى العديد من الاستنتاجات وتتلخص في التالي: 1. تخفيض تكلفة تأسيس اتصال من حيث المال بتقليل النسبة إلى 50% والوقت، زيادة تعقيد تخمين المفتاح، وكذلك الخوارزمية المستخدمة في الإرسال، وأخيرا إمكانية التحكم في طول المفتاح المناسب حسب مستوى الأمان المطلوب.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Glossaries

1. CIA Confidantiality, Integrety, Avaliability
2. VPN Vertual public network
3. GSM Global System for Mobile communications
4. LAN Local Area network
5. IO Input/Output
6. SMS Short Message Services
7. OpenSSL Open security soket layer
8. VDO Vanishing data object
9. PRNG Pseudo random number generator
10. AES Advanced Encryption Standard
11. R, P, N, Q Random number R which should be relatively prime to P and N. The numbers P and Q are prime numbers unique for each user
12. OTP One Time password
13. APK Advanced Port Knocking Authentication Scheme ALgorithm
14. KDOGSM Key agreement over GSM ALgorithm
15. WAN Wide Area network
16. QRC Quadratic Residue Cipher
17. PGP Pretty Good Privacy
18. PKI Public-key infrastructure
19. ECC Elliptic Curve Cryptography
20. DES Data Encryption Standard
21. MYSTY1 Block cipher algorithm
22. RC5 Rivest Cipher algorithm
23. TWOFISH Symmetric key block cipher with a block size of 128 bits and key sizes up to 256 bits

24. XXTEA Corrected Block TEA is a block cipher designed to correct weaknesses in the original Block TEA.

25. GOST Soviet and Russian government standard symmetric key block cipher

26. IV Initialization vector

27. ECB Modes of symmetric cryptography

28. CBC Modes of symmetric cryptography

29. OFB Modes of symmetric cryptography

30. CFB Modes of symmetric cryptography

31. EXE Executable file

32. LCD Liquid-crystal display

33. TOTP Time-based One-time Password Algorithm

34. CA Certification authority

35. KDS Key distribution center

36. RGB Red Green Blue Color

37. RMS Root Mean Square

38. Rijindael Pronounced rain-dahl is the algorithm that has been selected by the U.S. National Institute of Standards and Technology (NIST) as the candidate for the Advanced Encryption Standard (AES)

39. 3DES Triple Data Encryption Algorithm for symmetric-key block cipher,

40. RC2 Symmetric-key block cipher designed by Ron Rivest in 1987

41. MPA Multiple- packet knocking

42. IDEA International Data Encryption Algorithm

43. DHK Diffie-Hellmen key

44. MD5 Hash Generator function

45. SK1 Secrete Key 1

46. PA First communicator

47. PB Second communicator

48.	Tconnect	Start Communication timestamp
49.	Tsend	Timestamp of sending
50.	a,b	Random number
51.	g,p	Share public random
52.	PAS	Diffie-Hellmen PA to send PB to calculate the DHK
53.	PBS	Diffie-Hellmen PB to send PA to calculate the DHK
54.	SK2	Secrete key 2
55.	GUI	Graphical user interface

1 Chapter 1: Introduction

This chapter introduces the research project by a brief highlighting of the research problem, current solutions, and the proposed technique.

1.1 Sniffing Attacks

Data transfer over networks is considered as the best technique that improves the rate of achievements over distances in an immediate way. Although the rate of data transfer is a good point, there are many problems that may occur during the transfer process. One of these problems is called sniffing which has direct inconsistent with CIA triangle [41] in security principles. Sniffing of data removes the confidentiality of the data sent from the sender to the receiver in case of valuable or sensitive data. There are many studies that stand to defeat such problem or at least prevent the attackers from knowing what the parties transfer over the networks. Sniffing problem occurs because of the nature of the transfer process itself where the data moves from one node to another until it reaches the target machine. This mechanism allows the nodes to make a copy of the packets and read it in an unauthorized way. After many studies the only solution is to encrypt data in a way that the communication parties can decrypt it easily by agreeing on an encryption method. There are many encryption approaches to ensure that transferred data is understood by the parties. These approaches are: third party and key agreement. [3] [4] [5] [6] [7] [8] [9] [31] [32].

In information and communication technology, security of information is provided with cryptography. In cryptography, the key management is an important part of the whole system where the security lies on secrecy of cryptographic key. If the key size is large, the corresponding cryptographic algorithm ensures a guaranteed secure communication [1]. Distribution of the secret key is the main challenge in symmetric cryptography [2]. Key distribution techniques are classified based on network types as local area, wide area and threshold area conditions [1], now there is a general opinion to use the cryptography to ensure the data confidentiality according to CIA triangle.

1.2 Current solution approaches:

Many solutions for preventing data sniffing are proposed based on encryption of the data before sending it over the network. The data should be encrypted by using keys which are agreed between the communication parties before communication. The sender encrypts the data in an unreadable way by using the keys which are agreed on before. Encryption concepts are accepted widely in protecting systems and the data transferred over the networks. There are two approaches in the encryption field: Encryption based on third party which responses for encrypting data before sending from client to server. The other approach is based on encryption by key agreement between the data transfer parties, before communication establishes. Both approaches can guarantee that any attacker can sniff the data would not make use of it, unless getting the encryption key by stealing or guessing it.

1.2.1 Virtual public network VPN:

Extends a private network across a public network, such as the Internet. It enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network, and thus benefit from the functionality, security and management policies of the private network.

Figure 1-1 Show the VPN Architecture Shows that I have my virtual network, between the client and the VPN server this channel in red is secure, so I can send whatever I want in secure manner and no one can get the data transferred between the client and VPN server, but when I send data to any other client in the LAN the data, I will suffer from the man in the middle (Sniffing problem).

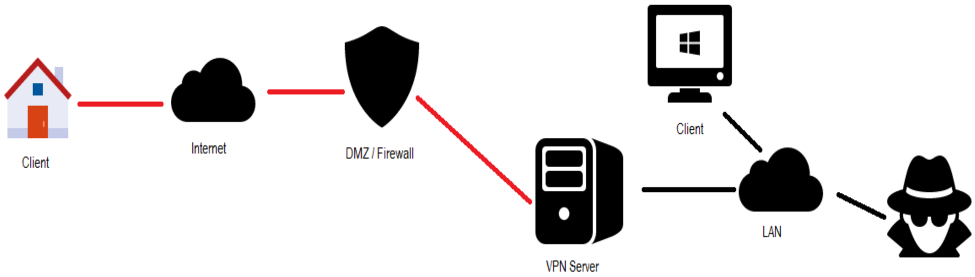


Figure 1-1 VPN Architecture Over WAN

1.2.2 Third-party approach and risks

Many researches and algorithms have been published and developed to avoid sniffing and stealing data or unauthorized data access. Such researches and algorithms based on two approaches: the first approach is based on third party to encrypt and secure data before, during and after transmission. Third party approach is getting widespread but not all data can be secured by third party, because of its sensitivity and there is no guarantee that data cannot be read by the third party itself, either by the organization itself or an employee in the it, because the organization is the one who is responsible for generating the key for encryption and decryption. There is a bug that has been identified called heart bleed which allows attackers to know information should be secured. Such bugs drive in combination with considering the third party as untrusted, to search for alternative solutions to avoid depending on third party and do the cryptography process ourselves.

Third-party risks examples:

As known, encryption/decryption stand on the basic of secure keys agreement between communication parties. Some threats are shown in [4] [5] [6] in the third party certificate especially those which are used in non-web base application as those in PayPal, Amazon and FBS. The data which has been transferred in previous examples was sniffed because of the bugs in the third party certification [4] [6]. Encryption always requires an encryption key to be agreed before communication between the communication parties. Such rules are assessed by depending on a third party to arrange the key and the encryption/ decryption rules. The behavior itself is a threat as we mentioned before: 1.The certification bugs as Heart Bleed. 2. The personal and ethical behavior of the third party or its employer.

Sniffing on the certification which used to secure data can be assessed in non-web based application[4][5][6], The validity of the safety of the third party certification in non-browsing application was not implemented correctly, which allows the man in the middle to sniff data and get benefit of the bug in the OpenSSL library, where the key and other login information can be sniffed, and that is clearly the opposite of what the researches try to assess to secure the data and information during the transmission process.

1.2.3 Key agreement approach and risks

In the second approach, researches and algorithms were developed and applied using the same idea, Key agreement before transmission and communication. The second approach mentioned before, is based on the parties must agree on an encryption key, the data is encrypted before sending and when receiving the data is decrypted to make use of it.

S. Barman [3] research based on finger print for each user can access the system where Biometric template is used to generate cryptographic key using key generation algorithm. In most of the cases, hash function is used to generate a stable binary string from biometric template. These approach completely inconsistence with Physical Security measurement because the finger print can be stolen from anything the authorized person touch.

C. Sur [7] research based on bit processing to encrypt the text by modifying an algorithm which responsible for bit rearrangement, C. Sur mentioned that when a file system is taken into consideration like a .doc, .pdf, .txt file is considered it will possess lots of letters inside and it is here that each letter can go through the algorithm. Such solution is a high processing cost for the parties; it makes a high delay in transmitting and that not valid for valuable data in short life time.

F.Yue. et al [8], research based on closed group and automatic self-destructing scheme after predefined period of time, but he mentions in his assumption that “No attacks on a VDO before it expires”, this assumption is not guarantee for the internet connection which mentioned in his assumptions.

V.Srivastava. et al [9], research based on that process is initiated by user by sending an SMS to the SMS Server connected to the PRNG requesting for the 256-bit One-Time Key which will act as the key for AES encryption and an 8-bit random number R which should be relatively prime to P and N. The numbers P and Q are prime numbers unique for each user and are already stored in the database server connected to the SMS server for each user. The only precursor for this process is that the user’s mobile number should be registered beforehand with the SMS Server. Request OTP <Allotted User ID><Fixed User Passcode><source IP-Address>the OTP is generated with validity time period so that there is no duplicate OTP generated before the use of original OTP. V.Srivastava. et al mention that the OTP will be send as a plan text on GSM which may but it under attacks [6].

1.3 Statement of the Problem

The main problem which rises after reviewing previous researches is the key distribution and how to protect it. The primary shortcoming of the APK v1.0 algorithm reported in [9], is that the distribution of the private key takes place in a plain text format over GSM network. This is a clear security risk since it can be easily exposed through sniffing attacks. This is a sever breach of the CIA triangle from the confidentiality aspect.

1.4 Objectives

1.4.1 Main Objective

The main objective of this research work is to overcome the private key plain text distribution vulnerability of the APK v.10 algorithm by implementing a modified version of the algorithm that ensures a protected exchange of the private key.

1.4.2 Specific Objectives

The specific objectives of the project are:

1. Analyze the current solution which concern with distribute the key over network to detect the shortage and overcome the detected shortage.
2. Design an algorithm to distribute the key over GSM securely.
3. Realize the algorithm and apply through government LAN, GSM Gate.
4. Evaluate the algorithm using a set of experiments, this experiments focuses on the measurement of financial cost, time cost, finally overall comparing of the proposed solution KDOGSM v1.0 and APK v1.0.

1.5 Importance

When we talk about sensitive data that should be restricted for a specific area, we have to encrypt it as mentioned before. Most application as banks, government and military systems are based on high secure type of transferring data. The APK v1.0 in [9] is secured but not enough. V.Srivastava. et al [9] suggests that the key will be sent over GSM but the breach here is that the key is going to be sent as a clear text, moreover he suggests to use a single algorithm AES which means if I sniff the key from the GSM, then I can easily read the content of the messages transferred between the two parties on the LAN.

Our attend is based on encrypting the key before distributing to the parties to avoid the sniffing of the key, this idea makes it difficult to read the key. On other hand it makes the sniffed packets over LAN hard to be understood.

1.6 Scope and Limitations

Our solution is able to encrypt/decrypt text over LAN with a technique based on calculated randomization, OTP (One Time Password) per connection, The randomization especially in key generation using one time password to overcome the static keys, send the key clearly over the same communication channel and send the key by clear text via GSM. Our solution is based on random selecting of encryption algorithms order, random communication parties numbering and sending encrypted key via SMS on GSM to secure the data which will be transferred over the communication channel (LAN).

Our work has limitations, which is not valid for all purposes. The limitations are

1. It works on LANs and WANs, but our works limited to LAN.
2. It is not valid for chatting for a long time, because if the attacker got the key, it’s easy to sniff the packets and decrypt it by guessing one of the five selected algorithms.
3. It is used for short term valued data because the time required to guess the encryption key and the encryption algorithms around days in average.
4. It transfers all data, but our model apply the text data only.
5. Smartphone should be connected during the whole session of communication.

1.7 Methodology

We will follow a research methodology that consists of the following phases:

1. Solution Design
    a. Specify the type of data to be sent.
    b. Select proper algorithms which are effective, secure, and less time cost.
    c. Consider the delay of the receiving SMS from GSM network
    d. Design the algorithms for generating the random numbers and keys for parties per connection.

2. Solution Implementation

    a. Build a desktop application that implements the proposed approach.

    b. Create a real environment to check the robustness of the mechanism.

    c. Built a desktop application that implements reversing our algorithm to state the trials of the key Guessing.

3. Solution Performance Evaluating

    a. Analyze our results of the implementation to evaluate it and prove that our consideration and constraints have been achieved correctly.

    b. Compare our performance results with the existing algorithms performance.

        i. Optimal key length.

        ii. Trials need to guess the key according to key length.

        iii. Financial cost of connection establishment.

1.8 Thesis Structure

Chapter 1 Introduction: Introduces the research project by a brief highlighting of the research problem, current solutions, and the proposed technique. Chapter 2 Technical and Theoretical Foundation: Describes the technical foundations needed for thesis work, including Data transfer, cryptography, third party, Terminology of symmetric cryptography algorithms and symmetric cryptography mode. Chapter 3 Related Works: Study and investigate methods and algorithms used for distribution key over LAN, and show how the network environment may affect the process of key transfer or running of applications. Chapter 4 Design and System Model: Discusses the background concepts of the “Advanced Port Knocking Authentication Scheme with QRC using AES” algorithm; the structural model that will be implemented, the general steps of the algorithm, the key generation algorithm, the encryption algorithm, the decryption algorithm, and the resulting enhancement on the whole algorithm. Chapter 5 Experiments and Results: Discuss the experimental design for our research work and the experiments results. Chapter 6 Conclusions and Future Directions: Present and discuss the conclusions for this work and the future directions

.

2 Chapter 2: Technical and Theoretical Foundation

This chapter describes the technical foundations needed for thesis work, including Data transfer, cryptography, third party, Terminology of symmetric cryptography algorithms and symmetric cryptography mode.

2.1 Introduction

Cryptography is the study of the mathematical techniques related to aspects of Information security such as confidentiality, data integrity and authentication. The basic idea of cryptography is to transmit information over an insecure channel and ensure that no entity in the middle can understand what is being transmitted. The history of cryptography goes back all the way to the Egyptians about 4000 years ago. It was also used extensively during the world wars [10]. As an illustration we can consider two people, say Alice and Bob, who wish to transmit information back and forth in such a way that a third person Oscar who is an opponent and can listen to the transmission but cannot decipher the information being transmitted. The information Alice intends to send is called the plaintext since it is not encrypted yet. Alice encrypts the plaintext with a predetermined key called the cipher key and transmits the cipher text over the insecure channel to Bob. Bob knowing the cipher key can decipher the cipher text to obtain the plaintext. Oscar, who does not have the cipher key, cannot decode the cipher text [11]. Figure 2-1 illustrates this.

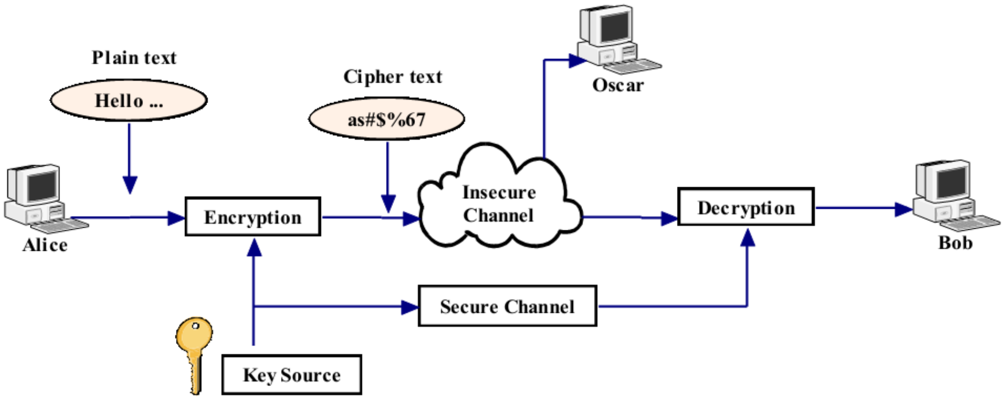


Figure 2-1 Cryptography System

Cryptography is probably the most important aspect of communications security and is becoming increasingly important as a basic building block for computer security [12].The increased use of computer and communications systems by industry has increased the risk of theft of proprietary information. Although these threats may require a variety of countermeasures, encryption is a primary method of protecting valuable electronic information [13].

Modern cryptography can be divided into two main subfields of study: Symmetric-key and Asymmetric-key cryptography. Symmetric-key can be divided into block ciphers and stream ciphers. Figure 2.2 depicts the taxonomy of cryptography.

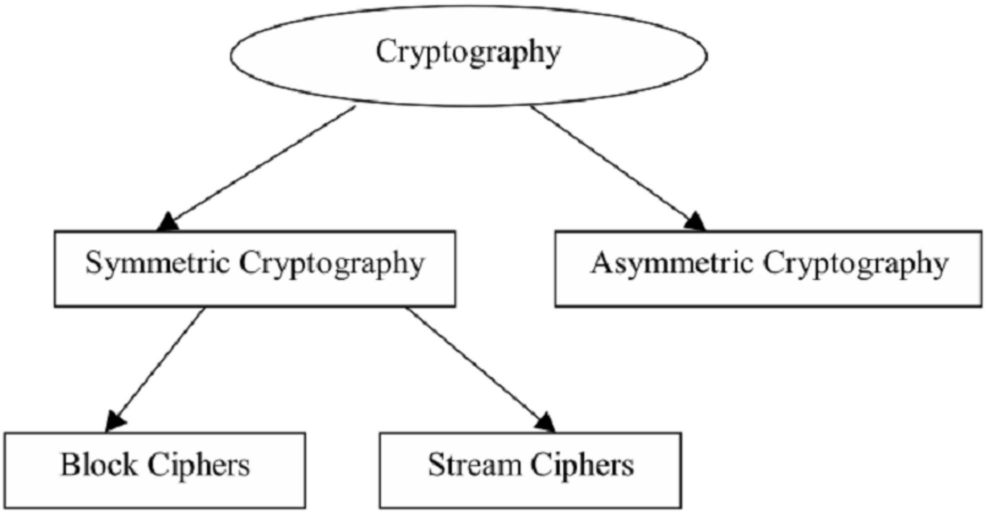


Figure 2-2 Taxonomy of Cryptography

2.2 Symmetric-key Cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976 [15].

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and their applications. We will discuss the terminology involved in symmetric-key cryptography in sections 2.3 and 2.4.

2.3 Asymmetric-key Cryptography

Asymmetric-key cryptography is also known as public-key cryptography where by two different but mathematically related keys are used a public key and a private key [16]. A public-key system is so constructed that calculation of one key (the 'private key') from the other (the 'public key') is computationally infeasible, even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair [17]. The public key may be freely distributed, while its paired private key must remain secret. The public key is typically used for encryption, while the private or secret key is used for decryption. The most famous applications of public-key cryptography are Elliptic-curve cryptography, PGP and the public-key infrastructure (PKI). Elliptic Curve Cryptography (ECC) provides the highest strength-per-key-bit of any cryptography algorithm known to take. Compared with other public-key approaches, ECC not only has the higher security but also has lows computation overhead, shorter key size and narrower bandwidth. Therefore, the experts believe that ECC will become the next generation widely used public-key cryptography.

Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting e-mails to increase the security of e-mail communication.

A PKI (public key infrastructure) enables users of a basically unsecured public network such as the Internet to securely and privately exchange data and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. PKI provides single sign on where a single login to the infrastructure system enables them to use that authentication token or service transparently through the rest of the infrastructure [11].

2.4 Cryptography Goals

There are five main goals of cryptography. Every security system must provide a bundle of security functions that can assure the secrecy of the system. These functions are usually referred to as the goals of the security system. These goals can be listed under the following five main categories [18]:

1. **Authentication**: The process of proving one's identity. This means that before sending and receiving data using the system, the receiver and sender identity should be verified.

2. **Privacy/confidentiality**: Ensuring that no one can read the message except the intended receiver. Usually this function is how most people identify a secure system. It means that only the authenticated people are able to interpret the message content and no one else.

3. **Integrity**: Assuring the receiver that the received message has not been altered in any way from the original. The basic form of integrity is packet check sum in IPv4 packets.

4. **Non-repudiation**: A mechanism to prove that the sender really sent this message. Means that neither the sender nor the receiver can falsely deny that they have sent a certain message.

5. **Service Reliability and Availability**: Since secure systems usually get attacked by intruders, which may affect their availability and type of service to their users. Such systems provide a way to grant their users the quality of service they expect.

2.5 Terminology of Symmetric Cryptography Algorithms

The terminology of symmetric cryptography algorithms mainly includes the following: **Plaintext**: Plain text is the ordinary information which the sender wishes to transmit to the receiver(s).

Ciphertext: The encrypted text is called Ciphertext. Encryption and Decryption: Encryption is the process of converting plain text into ciphertext as shown in Figure 2.3. Decryption is the reverse process, moving from ciphertext back to the original plaintext.

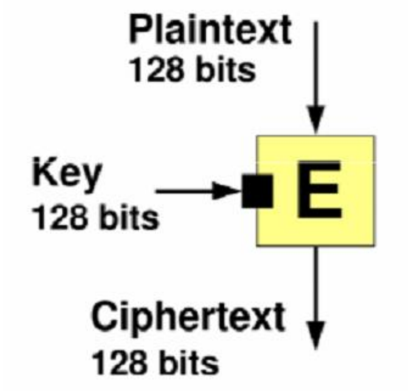


Figure 2-3A Typical Encryption Procedure

Cipher: A cipher is a pair of algorithms which ensure the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and by a specific key.

Key: The key is a secret parameter for encrypting or decrypting a specific message exchange context. Keys are important, as ciphers without keys are trivially breakable and therefore less than useful for most purposes.

Block Ciphers: The block cipher is a type of symmetric-key encryption algorithm that transforms a fixed-length block of plain text data into a block of ciphertext data of the same length. This transformation takes place under the action of a user provided secret key. Decryption is performed by applying the reverse transformation to the ciphertext block using the same secret key. The fixed length is called the block size and, for many block ciphers, the block size is 64 bits. In the coming years, the block size will increase to 128 bits as processors become more sophisticated. Since messages are almost always longer than a single block, some method of knitting together successive blocks is required. The different ways of knitting together blocks are known as the modes of operation and must be carefully considered when using block ciphers. Mode of operation will be further elaborated in section 4. Block ciphers can be easier to implement in software than stream ciphers, because they often avoid time consuming bit manipulations and they operate on data in computer-sized blocks [19].

Stream Ciphers: A stream cipher is a symmetric-key cipher where plaintext bits are combined with a pseudo-random cipher bit-stream (key stream), typically by an exclusive-or (xor) operation. In a stream cipher the plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption. Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity [19]. Stream ciphers that only encrypt and decrypt data one bit at a time are not really suitable for software implementation [19]. This explains why stream ciphers can be better implemented in hardware than block ciphers.

Feistel - Cipher: A Feistel cipher is a symmetric structure used in the construction of block ciphers; it is also commonly known as a Feistel network. A large proportion of block ciphers use the scheme, including Blowfish, DES, MYSTY1, RC5, TWOFISH, XXTEA, GOST, etc. The Feistel structure has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only a reversal of the key schedule. Therefore, the size of the code or the circuitry required to implement such a cipher is nearly halved [14]. Feistel networks combine multiple rounds of repeated operations, such as:

1. Bit-shuffling functions (often called permutation boxes or P-boxes).
2. Simple, non-linear functions (often called substitution boxes or S-boxes).
3. Linear mixing (in the sense of modular algebra) using XOR operations.

Key Size: key size or key length is the size of the key used in a given cryptographic algorithm. An algorithm's key length is distinct from its cryptographic security, which is a logarithmic measure of the fastest known computational attack on the algorithm, also measured in bits [20].

Rounds: Rounds is the number of iterations in a cipher system. From Figure 2.3 we can have a clear view that each repeated operations stand for a round. According to the crypto analysts, the bigger the number of rounds, the more secure the algorithms will be. The downside is that the execution time of the algorithms increases enormously.

2.6 Symmetric cryptography Modes

In cryptography, a mode of operation is an algorithm that uses a block cipher to provide an information service such as confidentiality or authenticity [21]. A block cipher by itself is only suitable for the secure cryptographic transformation (encryption or decryption) of one fixed-length group of bits called a block [22]. A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block [23] [24] [25].

Most modes require a unique binary sequence, often called an initialization vector (IV), for each encryption operation. The IV has to be non-repeating and, for some modes, random as well. The initialization vector is used to ensure distinct cipher texts are produced even when the same plaintext is encrypted multiple times independently with the same key [26]. Block ciphers have one or more block size(s), but during transformation the block size is always fixed. Block cipher modes operate on whole blocks and require that the last part of the data be padded to a full block if it is smaller than the current block size [22]. There are, however, modes that do not require padding because they effectively use a block cipher as a stream cipher.

There are several different modes of symmetric cryptography. ECB, CBC, OFB, and CFB are the earliest modes of operation that are have been defined where they are provide confidentiality, but they do not protect against accidental modification or malicious tampering [27].

2.7 Cryptanalysis

Cryptanalysis refers to the study of ciphers, cipher text, or cryptosystems (that is, to secret code systems) with a view to finding weaknesses in them that will permit retrieval of the plaintext from the cipher text, without necessarily knowing the key or the algorithm. This is known as breaking the cipher, cipher text, or cryptosystem [28]. Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single cipher text. There are two general approaches to attacking a conventional encryption scheme:

Cryptanalysis: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-cipher text pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

Brute-force attack: The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised. We first consider cryptanalysis and then discuss brute-force attacks. Table 2-1 summarizes the various types of cryptanalytic attacks, based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the cipher text only. In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the cipher text itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, an EXE file, a Java source listing, an accounting file, and so on.

Table 2-1 Types of Attacks on Encrypted Messages

Type of Attack	Known of Cryptanalyst
Ciphertext only	Encryption algorithm. Ciphertext to be decoded.
Known plaintext	Encryption algorithm. Ciphertext to be decoded. One or more plaintext-ciphertext pairs formed with the secret key.
Chosen plaintext	Encryption algorithm. Ciphertext to be decoded. Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with secret key.
Chosen ciphertext	Encryption algorithm Ciphertext to be decoded Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plain generated with the secret key
Chosen text	Encryption algorithm Ciphertext to be decoded Plaintext massage chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key. Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

The cipher text-only attack is the easiest to defend against because the opponent has the least amount of information to work with. Table 2-1 lists two other types of attack: chosen cipher text and chosen text. These are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack. Only relatively weak algorithms fail to withstand a cipher text-only attack. Generally, an encryption algorithm is designed to withstand a known-plaintext attack. Two more definitions are worthy of note. An encryption scheme is unconditionally secure if the cipher text generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much cipher text is available. That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the cipher text, simply because the required information is not there. With the exception of a scheme known as the one-time pad, there is no encryption algorithm that is unconditionally secure. Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

1. The cost of breaking the cipher exceeds the value of the encrypted information.
2. The time required to break the cipher exceeds the useful lifetime of the information.

An encryption scheme is said to be computationally secure if either of the foregoing two criteria are met. The rub is that it is very difficult to estimate the amount of effort required to cryptanalyze cipher text successfully.

A brute-force attack involves trying every possible key until an intelligible translation of the cipher text into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 2-2 shows how much time is involved for various key spaces. Results are shown for four binary key sizes.

Table 2-2 Average Time Required for Exhaustive Key Search

Key size (bit)	**Number of alternative keys**	**Time required at 1 decryption/ms**	**Time required at 106 decryption/ms**
32	232 =4.3 x 109	231 ms=35.8 minutes	2.15 milliseconds
56	256 =7.2 x 1016	255 ms=1142 years	10.01 hours
128	2128 =3.4 x 1038	2127 ms=5.4 x 1024 years	5.4 x 1018 years
168	2168 =3.7 x 1050	2167 ms=5.9 x 1036 years	5.9 x 1030 years
26 char. permutation	26! =4 x 1026	2 x 1026 ms=6.4 x 1012 years	6.4 x 106 years

2.8 One Time password (OTP)

OTP is a password that is valid for only one login session or transaction, on a computer system or other digital device [43]. OTPs avoid a number of shortcomings that are associated with traditional (static) password-based authentication.

2.8.1 Methods of generating the OTP

2.8.1.1 Time-synchronized

A time-synchronized OTP is usually related to a piece of hardware called a security token (e.g., each user is given a personal token that generates a one-time password). It might look like a small calculator or a keychain charm, with an LCD that shows a number that changes occasionally. Inside the token is an accurate clock that has been synchronized with the clock on the proprietary authentication server. On these OTP systems, time is an important part of the password algorithm, since the generation of new passwords is based on the current time rather than, or in addition to, the previous password or a secret key. This token may be a proprietary device, or a mobile phone or similar mobile device which runs software that is proprietary, freeware, or open-source. An example of time-synchronized OTP standard is Time-based One-time Password Algorithm (TOTP).

All of the methods of delivering the OTP below may use time-synchronization instead of algorithms.

2.8.1.2 Mathematical algorithms

Main article: Hash chain

Each new OTP may be created from the past OTPs used. An example of this type of algorithm, credited to Leslie Lamport, uses a one-way function (call it f). This one-time password system works as follows:

1. A seed (starting value) s is chosen.
2. A hash function f(s) is applied repeatedly (for example, 1000 times) to the seed, giving a value of: f(f(f(.... f(s) ....))). This value, which we will call f1000(s) is stored on the target system.
3. The user's first login uses a password p derived by applying f 999 times to the seed, that is, f999(s). The target system can authenticate that this is the correct password, because f(p) is f1000(s), which is the value stored. The value stored is then replaced by p and the user is allowed to login.
4. The next login, must be accompanied by f998(s). Again, this can be validated because hashing it gives f999(s) which is p, the value stored after the previous login. Again, the new value replaces p and the user is authenticated.
5. This can be repeated another 997 times, each time the password will be f applied one fewer times, and is validated by checking that when hashed, it gives the value stored during the previous login. Hash functions are designed to be extremely hard to reverse, therefore an attacker would need to know the initial seed s to calculate the possible passwords, while the computer system can confirm the password on any given occasion is valid by checking that, when hashed, it gives the value previously used for login. If an indefinite series of passwords is wanted, a new seed value can be chosen after the set for s is exhausted.

To get the next password in the series from the previous passwords, one needs to find a way of calculating the inverse function f−1. Since f was chosen to be one-way, this is extremely difficult to do. If f is a cryptographic hash function, which is generally the case, it is (so far as is known) a computationally infeasible task. An intruder who happens to see a one-time password may have access for one time period or login, but it becomes useless once that period expires. The S/KEY one-time password system and its derivative OTP are based on Lamport's scheme.

In some mathematical algorithm schemes, it is possible for the user to provide the server with a static key for use as an encryption key, by only sending a one-time password.

The use of challenge-response one-time passwords requires a user to provide a response to a challenge. For example, this can be done by inputting the value that the token has generated into the token itself. To avoid duplicates, an additional counter is usually involved, so if one happens to get the same challenge twice, this still results in different one-time passwords. However, the computation does not usually involve the previous one-time password; that is, usually this or another algorithm is used, rather than using both algorithms [43].

2.8.1.3 Several techniques that use OTP [45]:

3. Lin OTP: is acronym for Linux One Time Password that uses OTP to increase the security of all types of logon processes.

4. MOTP: is acronym for Mobil One Time Password which deals with synchronization between client and server with period of time usually 3munities; several software downloaded on mobiles support this technology.

5. SMSOTP: SMS OTPs are used as an additional factor in a multi-factor authentication system. Users are required to enter an OTP after logging in with a user name and password.

6. HOTP: is acronym for HMACOne Time Password algorithm based on an increase in counter value. Both client and server have a counter value. Server generates the password for using the counter. If both passwords match, the server authenticates the user and updates the counter (increment/ decrement the counter).

7. CROTP: is acronym for The OATH Challenge-Response algorithm based on challenge from authentication server. While server sends random challenge consists of 4 character defined as PIN, the user enters PIN value then sends response to the server.

8. TOTP: is acronym for Time One Time Password is used as an additional factor in a two factor authentication system. Users are required to enter an OTP after logging in with a user name and password to generate OTP in a period of time.

2.8.2 OTP Comparison

Implementing and analyzing three OTP techniques to prevent replay attacks in RADIUS protocol and provided a comparison between these techniques of our ELSBOT by considering a set of factors such as preventing replay attack, CPU overhead, technique speed, server response time and OTP duration. After testing and evaluating of our work, the researchers found that the three OTP techniques prevent the replay attack in RADIUS environment. The CPU overhead at TOTP technique is less than others and the speed at TOTP technique is the highest while the speed at CROTP technique is higher than HOTP technique. The average server response time at TOTP technique in our ELSBOT is the best in terms of server response time. Finally, according to these results, we reach that the TOTP is the most secure technique because this OTP is valid for a short time, while the CROTP is a more secure than HOTP because the server challenges us with the random PIN in the CROTP. Thus, our ELSBOT is an efficient overall solution from the security perspective and it will be much more difficult for attackers to reach the ELSBOT server [45]

2.9 Virtual Private Network

A VPN is a private network that uses a public infrastructure (usually the Internet) to connect remote sites or users [44]. The VPN as the name suggest uses “virtual “connections routed through the Internet from the business's private network to the remote site or remote employee. It is a new technology which can be applied to LAN as well as to WLAN.

A VPN maintains privacy of data through security procedures and tunneling protocols. In effect, data is encrypted at sender’s side and forwarded via "tunnel” which is then decrypted at receiver’s side. An additional layer of security can be added by encrypting not only the data, but also the originating and receiving network addresses. Two VPN technologies that are being used are:

1. **Site-to-site VPN** - A site-to-site VPN allows multiple offices in fixed locations to establish secure connections with each other over a public network such as the Internet. It also provides extensibility to resources by making them available to employees at other locations.
2. **Remote Access VPN** - A remote-access VPN allows individual users to establish secure connections with a remote computer network. These users can access the secure resources on that network as if they were directly plugged in to the network's servers.

2.9.1 Features in VPN

1. Provide extended connections across multiple geographic locations without using a leased line.

2. Improved security mechanism for data by using encryption techniques.
3. Provides flexibility for remote offices and employees to use the business intranet over an existing Internet connection as if they're directly connected to the network
4. Saves time and expense for employees who commute from virtual workplaces
5. VPN is preferred over leased line since leases are expensive, and as the distance between offices increases, the cost of leased line increase.
6. IPSec VPN and SSL VPN are two solutions of VPN which are widely used in WLAN. We will discuss both of them together with their advantages and disadvantages.

2.9.2 Disadvantages of IPSec and SSL VPN

1. To establish a secure connection using IPSec VPN, a VPN Client is needed to be configured and installed on every terminal for data transmission.
2. Installation and management of VPN client on every machine leads to expenditure which consequently increases with growing number of mobile users.
3. IPSec VPN operation requires specialized training because of the software and hardware client installed.
4. SSL"s primary disadvantage is that it operates at application layer, limiting access to only those resources that are browser- accessible.
5. Requires Java or ActiveX downloads to facilitate access to non-Web-enabled applications.
6. SSL tunneling is not supported on Linux or non-Windows operating systems.

2.10 Conclusion

As this chapter has shown the background concepts of this work, not only does it give you the basic cryptography theory and cryptography goals, but it also provides more information about symmetric cryptography such as terminology of symmetric cryptography algorithms, symmetric cryptography modes and cryptanalysis. Guiding you along the ways is used to hacking the cryptography key to decrypt the cipher text that is transmitting between the sides of communication in the network.

3 Chapter 3: Related Works

In this chapter we study and investigate methods and algorithms used for distribution key over LAN, and show how the network environment may affect the process of key transfer or running of applications.

3.1 Third party approach

As known encryption/decryption stand on basic of key agreement for both communication parties. There is an approach in cryptography called CA certification authority which is known as trusted certificates, which are typically used to make secure connections to a server over the Internet. A certificate is required in order to avoid the case that a malicious party (Sniffers) which happens to be on the path to the target server pretends to be the target. Such a scenario is commonly referred to as a man-in-the-middle attack. The client uses the CA certificate to verify the CA signature on the server certificate, as part of the checks before establishing a secure connection. Certification suffered some threats [4] [5] [6] in the 3rd party certificate especially those which are used in non-web base application as those in PayPal, Amazon and FBS. The data been transferred using previous examples were sniffed because of the bugs in the 3rd party certification [4] [6]. Either a notable case of CA subversion like this occurred in 2001, when the certificate authority VeriSign issued two certificates to a person claiming to represent Microsoft. The certificates have the name "Microsoft Corporation", so they could be used to spoof someone into believing that updates to Microsoft software came from Microsoft when they actually did not. The fraud was detected in early 2001. Microsoft and VeriSign took steps to limit the impact of the problem. On other hand the certification also has a bugs as Heart Bleed [2], the personal and ethical behave of the third party or its employer, all this points makes person thinks thousand time before securing his data using third part.

3.2 Key agreement approach

Second approach is based on two parties agree on an encryption key, the data encrypt before send and when received the data decrypted to make use of it.

3.2.1 Finger print

S. Barman [3] research based on finger print for each user can access the system where Biometric template is used to generate cryptographic key using key generation algorithm. In most of the cases, hash function is used to generate a stable binary string from biometric template. These approach completely inconsistence with Physical Security measurement because the finger print can be stolen from anything the authorized person touch.

Algorithm phases:

Table 3-1 Cryptographic Key Distribution through Fingerprint

Phase	**Process**
Phase 1	Enrolment of users
Phase 2	Key Distribution Process

User's fingerprint data is enrolled to KDC. The fingerprint image is pre-processed and are extracted from fingerprint image. The fingerprint data is transformed in cancelable template with one way transformation function. User uses a random vector (P) as transformation parameter to provide diversity to the fingerprint data. The cancelable template is sent to KDC through a secure channel or it is submitted to KDC in offline basis. KDC assigns a unique user id to the registered user and stores the cancelable template along with user id in the database. The cancelable template will be used to generate permanent key of a user. The user must store the transformation parameter which is used at the time of cancelable template generation for enrolment. This key will be used to generate the same cancelable template from similar fingerprint using the same algorithm.

3.2.2 Bit Processing

C. Sur [7] research based on bit processing to encrypt the text by modifying an algorithm which responsible for bit rearrangement, C. Sur mentioned that When a file system is taken into consideration like a .doc, .pdf, .txt file is considered it will possess lots of letters inside and it is here that each letter can go through the algorithm. Such solution is a high processing cost for the parties; it makes a high delay in transmitting and that not valid for valuable data in short life time.

3.2.3 Randomized Cryptographic Key Generation Using Images

The authors in [31], proposed system focuses on generating key based on images. The generated key need not have to be stored. It can be generated anywhere using the image and the session. To generate the original message from the cipher text one has to know the session on which the encryption is done, image considered for encryption, RGB values taken and processed for key generation. The key generation is based on the image stored in the database. Consider the pixel value of an image and extract one channel (RGB) at a time. It can be either a red channel or green channel or blue channel. The values of the components should be stored in an array. The array size should be p*q where p and q are the resolution of an image. The key is generated based on the following steps. They considered all the channels for key generation.

1. **Red channel**: Consider only the diagonal values of the array whose indexes (N) are N%8=0. RMS(Root Mean Square) value for those diagonal values are calculated. The generated RMS value is considered as a part of the key and it is stored in a variable.
2. **Green channel**: Sum up the all the pixel values in a zigzag manner starting from 0x0 to (n-1) x (n-1), n=size of the image and store the summed up value in a variable.
3. **Blue channel**: Repeat the steps of the red channel. Now, the value got using the red channel is appended with the value got using the green channel and the blue channel value is also appended to generate the key. The sender encrypts the confidential message using the RC5 algorithm, once the encryption is done it is sent to the receiver along with the session log. The session log contains the time in which the encryption is done.

In reference to the session log the receiver will consider the image in the image database to generate key for decryption. The generated key along with the encrypted message is sent for decryption (RC5 Algorithm) and the original message is extracted.

3.2.4 Avoiding Key Exchange

In [32], new type of Symmetric encryption algorithms is introduced. This new symmetrical encryption algorithm is proposed to prevent the outside attacks. The new algorithm avoids key exchange between users and reduces the time taken for the encryption and decryption. It operates at high data rate in comparison with The Data Encryption Standard (DES), Triple DES (TDES), advanced encryption Standard (AES-256), and RC6 algorithms.

The new algorithm avoids fixed-key exchange between sender and receiver with each authentication process in WLAN.

The new algorithm uses a key size of 512-bits to encrypt a plaintext of 512-bits during the 16-rounds. In this Algorithm, a series of transformations have been used depending on S-BOX, different shift processes, XOR-Gate, and AND-Gate. The S-Box is used to map the input code to another code at the output.

The key generation generates 16-keys during 16-rounds. One key of them is used in one round of the encryption or decryption process. In the first time, the initial key is divided into four parts a, b, c, and d, 128-bits each. In each round of the key generation, there are series of the transformation used to generate the round-key.

The Encryption process in the new algorithm is used to encrypt the plaintext of size of 512-bits by a key of size of 512-bits in each round during 16-rounds. Series of transformations are applied on the plaintext in each round to obtain a ciphertext finally.

The key-updating is a new approach to increase the difficulty to discover the key. The text and speech signals are used to prove the success of the proposed algorithm. The proposed algorithm has higher data rate than DES, TDES, and AES algorithms. The voice encryption and decryption is applied using wired and wireless connection.

3.2.5 Other key agreement approaches

F.Yue. et al [8] research based on closed group and automatic self-destructing scheme after predefined period of time, but he mentions in his assumption that “No attacks on a VDO before it expires”, this assumption is not guaranteed for the connection which mentioned in his assumptions.

V.Srivastava. et al [9] research based on that process is initiated by user by sending an SMS to the SMS Server connected to the PRNG requesting for the 256-bit One-Time Key which will act as the key for AES encryption and an 8-bit random number R which should be relatively prime to P and N figure 3-1, figure 3-2. The numbers P and Q are prime numbers unique for each user and are already stored in the database server connected to the SMS server for each user. The only precursor for this process is that the user’s mobile number should be registered beforehand with the SMS Server. Request OTP <Allotted User ID><Fixed User Passcode><source IP-Address> the OTP is generated with validity time period so that there is no duplicate OTP generated before the use of original OTP.

V.Srivastava. et al mention that the OTP will be send as a plan text on GSM which may but it under attacks [6].

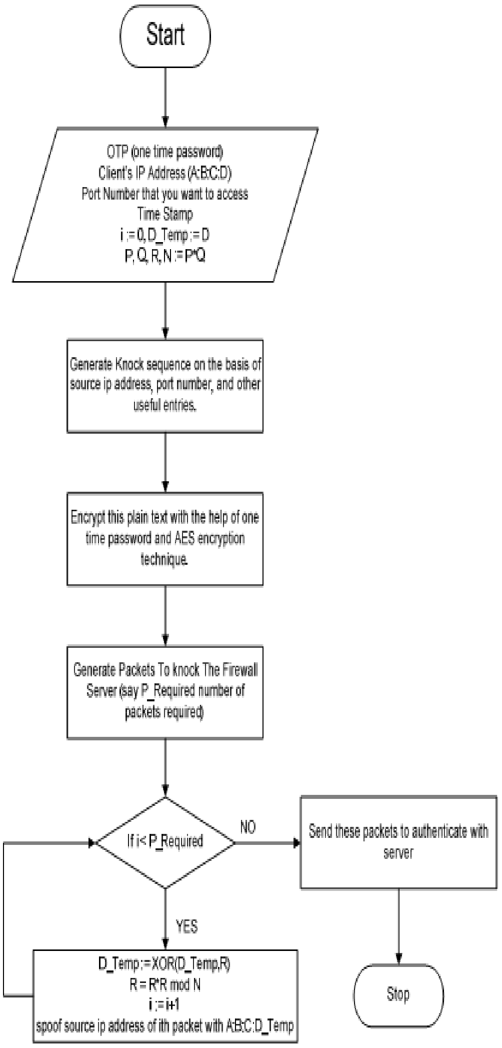

Start
OTP (one time password)
Client's IP Address (A:B:C:D)
Port Number that you want to access
Time Stamp
i := 0, D_Temp := D
P, Q, R, N := P*Q
Generate Knock sequence on the basis of source ip address, port number, and other useful entries.
Encrypt this plain text with the help of one time password and AES encryption technique.
Generate Packets To knock The Firewall Server (say P_Required number of packets required)
If i< P_Required
NO
Send these packets to authenticate with server
YES
D_Temp:=XOR(D_Temp,R)
R = R*R mod N
i := i+1
spoof source ip address of ith packet with A:B:C:D_Temp
Stop

Figure 3-1Flowchart of activity at the client side

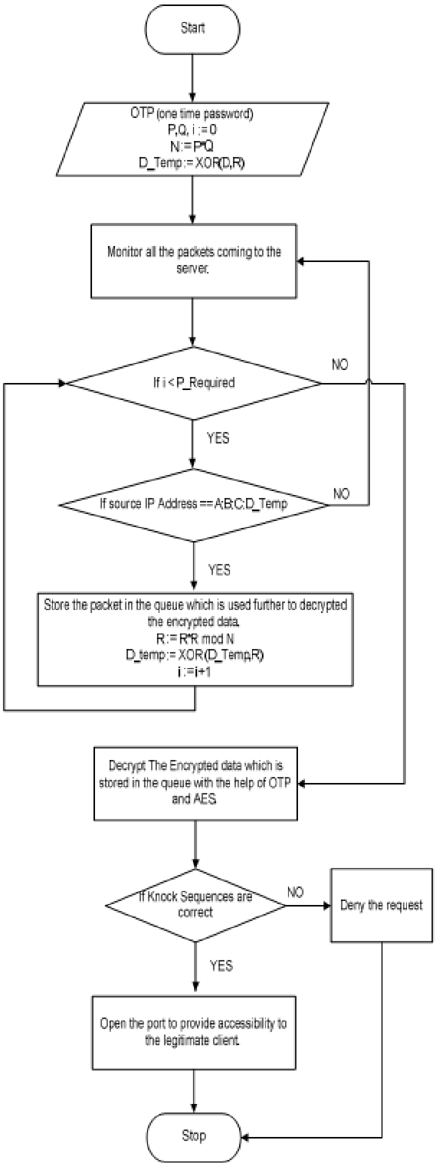
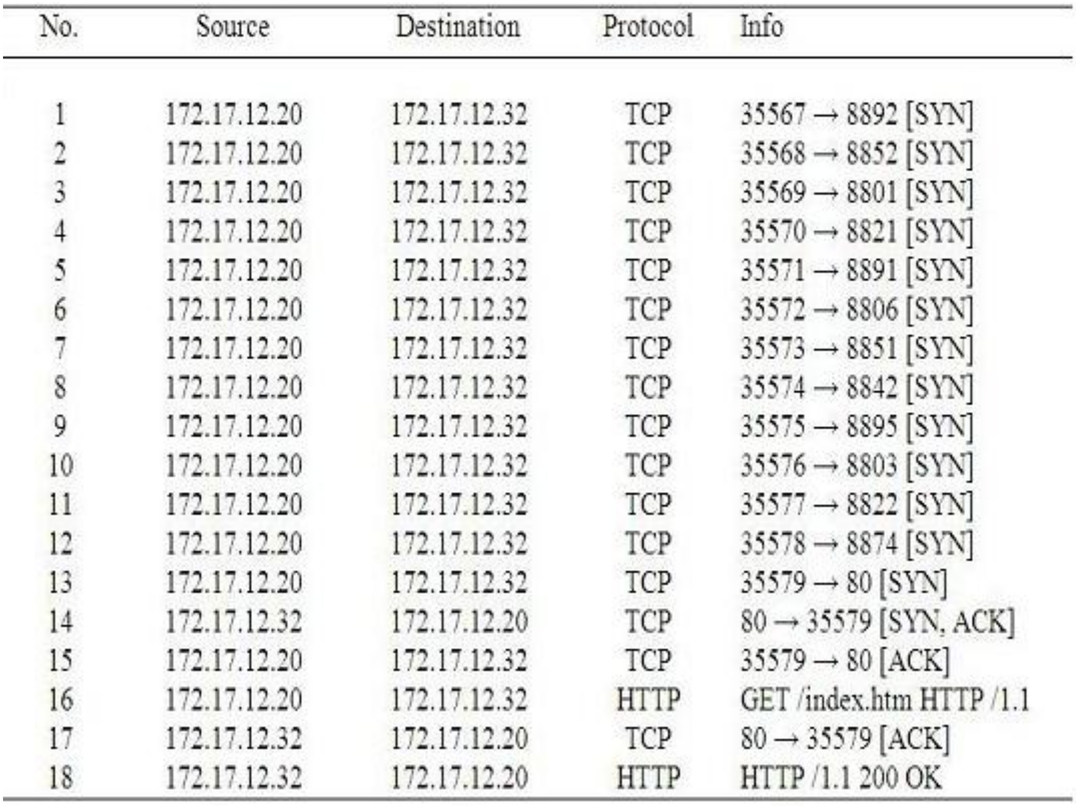

Start
OTP (one time password)
P,Q, i := 0
N := P*Q
D_Temp := XOR(D,R)
Monitor all the packets coming to the server.
If i < P_Required
NO
YES
If source IP Address == A:B:C:D_Temp
NO
YES
Store the packet in the queue which is used further to decrypted the encrypted data.
R := R*R mod N
D_temp := XOR(D_Temp,R)
i := i+1
Decrypt The Encrypted data which is stored in the queue with the help of OTP and AES.
If Knock Sequences are correct
NO
Deny the request
YES
Open the port to provide accessibility to the legitimate client.
Stop

Figure 3-2Flowchart of activity at server side

During the knocking process by the client Table 3-2, server offers no response and it just monitors the knocks silently on the specified ports. When the server detects a valid knock sequence, it triggers a server side process and opens the port for communication with the client. Such method either not valid for transmitting sensitive data neither urgent data. The numbers of request and SMS makes the suggestion hard to be reliable in case of sensitive data where the required bandwidth to achieve the required sequence makes a delay, either sending 2 SMS to establish a connection makes the solution costly if the operation is repeated over the 24 hours.

Table 3-2 Captured Data showing standard port knock sequence



No.	Source	Destination	Protocol	Info
1	172.17.12.20	172.17.12.32	TCP	35567 → 8892 [SYN]
2	172.17.12.20	172.17.12.32	TCP	35568 → 8852 [SYN]
3	172.17.12.20	172.17.12.32	TCP	35569 → 8801 [SYN]
4	172.17.12.20	172.17.12.32	TCP	35570 → 8821 [SYN]
5	172.17.12.20	172.17.12.32	TCP	35571 → 8891 [SYN]
6	172.17.12.20	172.17.12.32	TCP	35572 → 8806 [SYN]
7	172.17.12.20	172.17.12.32	TCP	35573 → 8851 [SYN]
8	172.17.12.20	172.17.12.32	TCP	35574 → 8842 [SYN]
9	172.17.12.20	172.17.12.32	TCP	35575 → 8895 [SYN]
10	172.17.12.20	172.17.12.32	TCP	35576 → 8803 [SYN]
11	172.17.12.20	172.17.12.32	TCP	35577 → 8822 [SYN]
12	172.17.12.20	172.17.12.32	TCP	35578 → 8874 [SYN]
13	172.17.12.20	172.17.12.32	TCP	35579 → 80 [SYN]
14	172.17.12.32	172.17.12.20	TCP	80 → 35579 [SYN, ACK]
15	172.17.12.20	172.17.12.32	TCP	35579 → 80 [ACK]
16	172.17.12.20	172.17.12.32	HTTP	GET /index.htm HTTP /1.1
17	172.17.12.32	172.17.12.20	TCP	80 → 35579 [ACK]
18	172.17.12.32	172.17.12.20	HTTP	HTTP /1.1 200 OK

3.3 Virtual public network VPN

Virtual public network (VPN) is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed though some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis.

VPN Main function is to provide a secure media to isolate the client of the WAN and put it in a LAN to do his business securely of the man WAN he connected to. But on other hand we talk about the security in the LAN we isolated in we can say that if there is a Man-in-Middle in this isolated LAN physically, he can sniff the packets easily where the content of theses packets is not encrypted, but the secured tunnel from outside WAN to isolated LAN is secure.

3.4 Conclusion

This chapter presented and discussed a number of related works in the field of key agreement, agreement and distribution. Based on the discussions above, it is clear that existing research works have the following set of shortcomings:

1. Heavy algorithm processing as author mention in his research paper based on single algorithm for bit processing in fix way, the algorithm is user defined.
2. Depending on human physical unique signature like finger print, which inconsistent with security physical measurement.
3. The assumption of the LAN network is secure when sending key.
4. Sending the key as plain text either on LAN or GSM.

In our research we focus on problem number 4. As a result of our work, some of the other problems have already been treated. This is presented in more details in the next chapter.

4 Chapter 4: Proposed Technique

This chapter discusses the background concepts of the “Advanced Port Knocking Authentication Scheme with QRC using AES” algorithm [9]; the structural model that will be implemented, the general steps of the algorithm, the key generation algorithm, the encryption algorithm, the decryption algorithm, and the resulting enhancement on the whole algorithm.

4.1 Introduction:

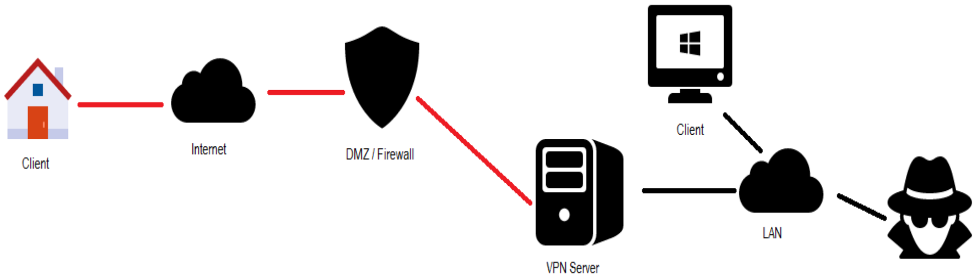
Our scenario is that two terminals need to exchange data through an unsecured channel. They need to ensure the confidentiality and authenticity of the messages transmitted between them, so they should use cryptography techniques to prohibit a third terminal from intercepting the exchanged data. They need to agree on the cryptography key before they can communicate with each other. Since the channel is unsecure to data exchange, it is unsecure to key exchange which should be secure and periodically changed to assure high security. Instead of exchanging session keys, we need a simple and secure method to generate the key in both sides. Advanced Port Knocking Authentication Scheme with QRC using AES is a new type of symmetric encryption algorithm that prevents the attacks and avoids key exchange between the two sides of communication by creating the key in home on the sender/receiver machine [9].

In this work, we have implemented the key distribution over GSM to a specific scenario that will be detailed in this chapter in order to measure the strength against the cryptanalysis attack and the efficiency of the algorithm against well-known symmetric crypto graphic algorithms which are AES, Rijindael, DES, 3DES and RC2. We decided to update this algorithm (Advanced Port Knocking Authentication Scheme with QRC using AES) to increase the strength and efficiency to be in the forefront, so during this chapter, we will explain the previous algorithm which will called “APK v.1". We will also explain the new enhanced algorithm which is modified from the previous algorithm. The new algorithm is called "KDOGSM v.1"

4.2 Virtual public network VPN

Virtual public network (VPN) is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed though some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis.

VPN Main function is to provide a secure media to isolate the client of the WAN and put it in a LAN to do his business securely of the man WAN he connected to. But on other hand we talk about the security in the LAN we isolated in we can say that if there is a Man-in-Middle in this isolated LAN physically, he can sniff the packets easily where the content of theses packets is not encrypted, but the secured tunnel from outside WAN to isolated LAN is secure as Figure 4-1.



4-1VPN Architecture over WAN and the reached LAN

4.3 System model of current APK v.1

The structure model of APK v.1 implementation approach in Figure 4.2 [9]. The block diagram shows that the host A and B are two hosts in the local area network, and they are communicating with each other by exchanging text messages. The model provides the two hosts a secure channel so they can exchange the data without doubting about the confidentiality and authenticity of the transmitted data over channel.

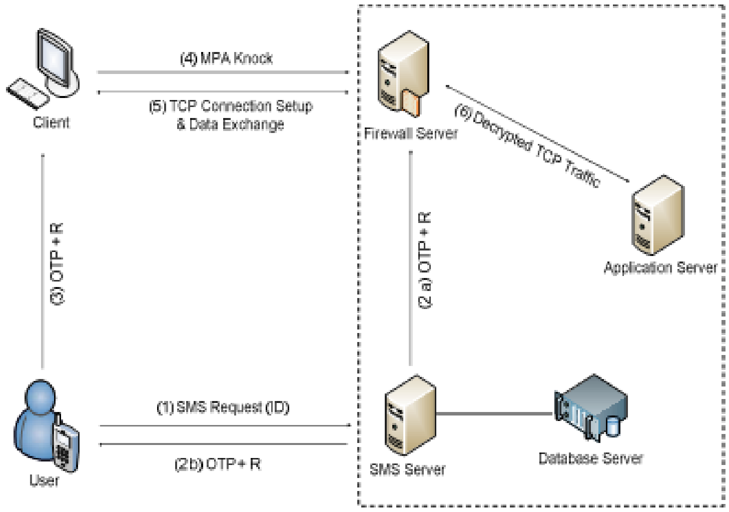


Figure 4-2Port Knocking Architecture [9]

4.3.1 APK work flow:

1. The process is initiated by user by sending an SMS to the SMS Server connected to the PRNG requesting for the 256-bit One-Time Key.
2. One-Time Key will act as the key for AES encryption
3. An 8-bit random number R which should be relatively prime to P and N.
4. The numbers P and Q are prime numbers unique for each user and are already stored in the database server connected to the SMS server for each user.

The SMS is sent through a dedicated channel which in out of band for general communication. The reply sent by the server is also send through the same dedicated channel. The only precursor for this process is that the user's mobile number should be registered beforehand with the SMS Server. The message send by the user should be of the form Request OTP <Allotted User ID><Fixed User Pass-code><source IP-Address>.The OTP is generated with validity time period so that there is no duplicate OTP generated before the use of original OTP or the expiration of time period. This is done to prevent the duplicity of OTP for the same user ID to prevent denial of service attacks on the PRNG server and SMS server. The random number R is of 8-bit which will be used to spoof the IP-address every time the request packet is sent to the application server through any firewall. The reply SMS from the SMS Server should be of the form [9] <Time Stamp><One Time Key><Random Number R> Time stamp is used to verify authenticity of the OTP and will be required in the prevention of spoofing of the mobile numbers and DOS attacks. The obtained OTP is then passed to the client to work as the key for the AES which will be used to encrypt the data to be sent i.e. the knock sequence which is our authentication data in this case and R which will be used in the Quadratic Residue Cipher.

4.3.2 APK v1.0 example:

Assume that source IP-Address is 172.17.12.20 and the value of p = 11, q = 17. Firewall server will randomly pick the value of R (here R = 7), which is relatively prime to N (=p*q). After the generation of knock sequences on the basis of source IP-Address, the destination port number and other entries are encrypted using AES and OTP. Once all the packets are generated to knock the firewall server, the algorithm follows the flowchart depicted in Figure 4.3. The source IP-Address has been spoofed of each packet that will be involved in multiple-packet knocking (MPA). If we XOR 20 and 7, we get 19. Therefore, the first source IP-Address is 172.17.12.19. The value of 72 mod (11*17) = 49 and XOR of 19 and 49 is 34. Therefore the second source IP-Address in the sequence is 172.17.12.34. This process is repeated and the following table is generated

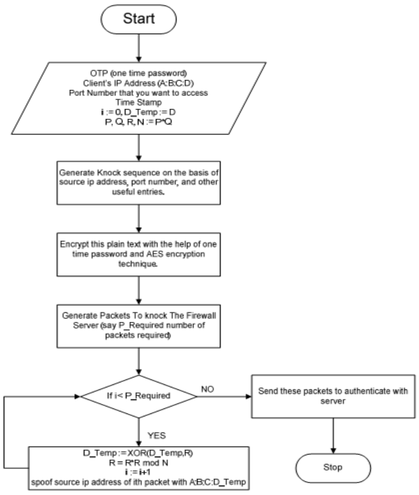

Start
OTP (one time password)
Client's IP Address (A:B:C:D)
Port Number that you want to access
Time Stamp
i := 0, D_Temp := D
P, Q, R, N := P*Q
Generate Knock sequence on the basis of source ip address, port number, and other useful entries.
Encrypt this plain text with the help of one time password and AES encryption technique.
Generate Packets To knock The Firewall Server (say P_Required number of packets required)
If i< P_Required
NO
Send these packets to authenticate with server
Stop
YES
D_Temp := XOR(D_Temp,R)
R = R*R mod N
i := i+1
spoof source ip address of ith packet with A:B:C:D_Temp

Figure 4-3Flowchart of activity at the client side [9]

4.4 System model for proposed APK (KDOGSM v.1)

The structure model of "KDOGSM v.1" implementation approach is depicted in Figure 4.4. From figure 4.4, we can conclude that the structure model of "KDOGSM v.1" workflow process.

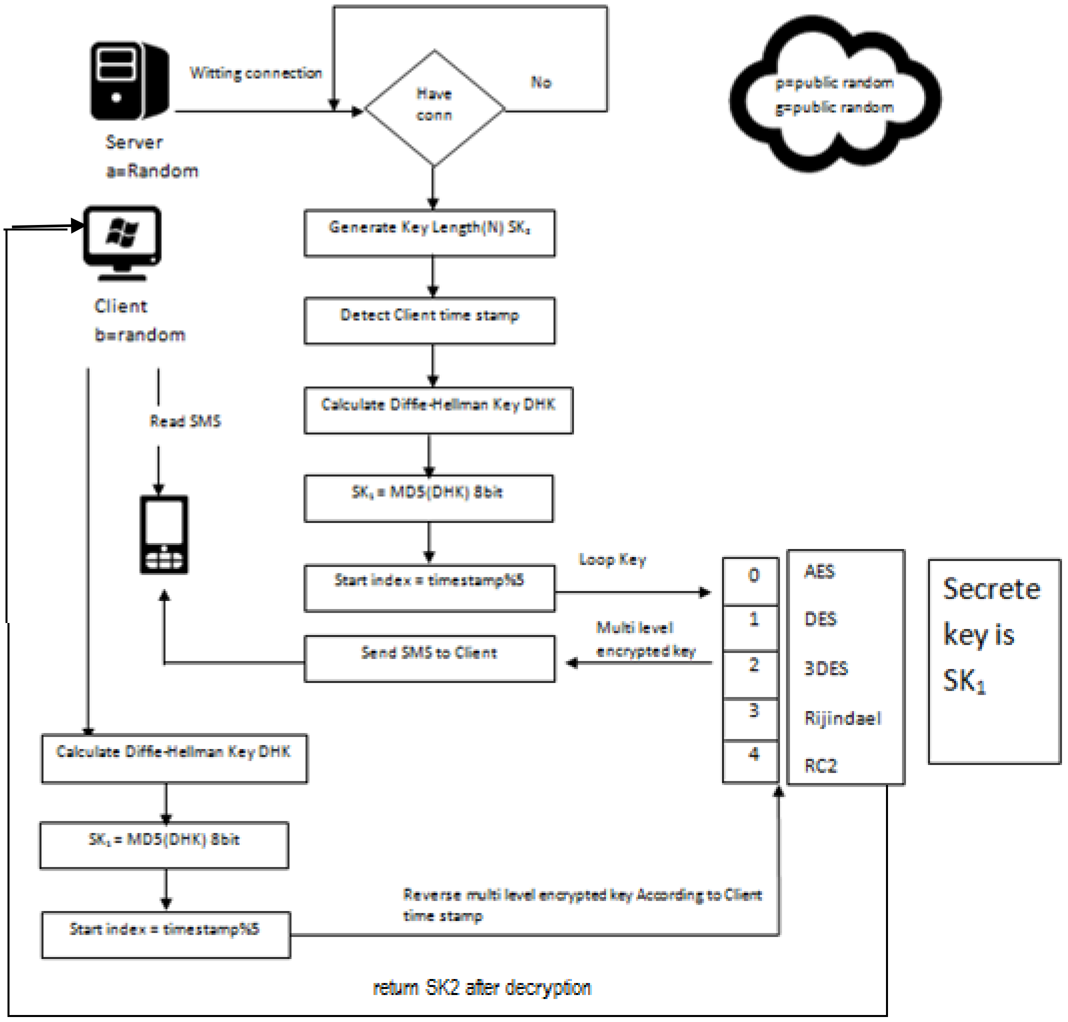

Witting connection
Have conn
No
p=public random
g=public random
Server
a=Random
Generate Key Length(N) SK1
Client
b=random
Detect Client time stamp
Calculate Diffie-Hellman Key DHK
Read SMS
SK1 = MD5(DHK) 8bit
Loop Key
Start index = timestamp%5
0
1
2
3
4
AES
DES
3DES
Rijindael
RC2
Secrete key is SK1
Multi level encrypted key
Send SMS to Client
Calculate Diffie-Hellman Key DHK
SK1 = MD5(DHK) 8bit
Reverse multi level encrypted key According to Client time stamp
Start index = timestamp%5
return SK2 after decryption

Figure 4-4Key distribution Over GSM

4.4.1 Multi-level encryption Idea:

Separation of key transfer network and the communication network takes place in order to increase security level. In our proposed techniques we send the key over GSM and the data over LAN. Before sending the key, we encrypt it in multi-level encryption algorithms. The idea of multi-level encryption is used to makes the brute-force attack possibilities large set than using single encryption algorithms, each decryption output from one level is considered as input for the next decryption process, until getting out the multi-level decryption process. Depending on this idea we make the number of guessing the key trials increased by multiplying this number by the count of the multi-level algorithms used in this proposed techniques.

4.4.2 Randomization selection of the encryption algorithms

The more complexity is when using single encryption algorithm, the attacker's main challenge is to guess the key by implementing the Brute-force attack on the used single algorithm, but in our technique, in each sending process we use a different algorithm based on sending time.

4.4.3 Predefined array of algorithms

Mansoor Ebrahim et al. [36]. After analyzing the most popular symmetric algorithms AES (Rijndael) was found the most secure, faster and better among all the existing algorithm with no serious weaknesses, there are some flaws in symmetric algorithms such as weak keys, insecure transmission of secret key, speed, flexibility, authentication and reliability i.e. in DES, four keys for which encryption is exactly the same as decryption [37]. This means that Original plain text can be recovered, if the encryption is applied twice with one of these weak keys [37]. DES is very slow when implemented in software; the algorithm is best suited to implementation in hardware. Similar is the case in IDEA that involves large class of weak keys facilitating the cryptanalysis for recovering the key. DES and IDEA have the same encryption speed on. Triple DES does not always provide the extra security that might be expected making use of double and triple encryption as well as it is very slow when implemented in software as it is derived from DES, and DES on software is already slow, so Triple-DES might be considered safest but slowest. In Blow Fish there are certain weak key that attacks its three-round version [36], further it is also exposed to a differential attack against its certain variants, it also slow in speed but much more faster than DES and IDEA. While looking at the five finalists of AES no serious weakness was found, however few feeble aspect was highlighted that might be exploit as a molest in near future, such as in AES (Rijndael) a numerical property of the cipher might be exposed into an attack, full RC6 arbitrariness is not achieved, Serpent a bit slower and complex, Twofish possibly suspected to chosen-key attacks and MARS relatively complex to analyze. [38] [39].

This selection not restricted because we can attach any symmetric encryption/ encryption algorithms of the well-known algorithms, or user defined algorithm. We have select the best algorithms in performance, security and memory usage according to [36], we provide the flexibility for the system runner to change whatever he want by customizing the setting as he wish.

The order of the algorithms is meaningless for this application. Because the start algorithm always set by the time not by the last one been chosen.

4.4.4 Why we use 2 key for this algorithm

According to figure 4.4 we have generate Diffie-Hellmen DHK then convert it to MD5 hash, because the DHK range always small integer because the calculation method which is modules used in the Diffie-Hellmen algorithms, so if we depend on this key as the main key it is going to be easy to guess that key.

In order to overcome this problem we have to convert that number to MD5 hash to generate the key of length N which make the algorithm stronger in face of the brute force attack by enlarging the dataset of key guessing.

4.4.5 Algorithm encryption/ decryption work flow

KDOGSM v.1 is based on encrypt the distributed key which will be used to encrypt the text over the communication channel. The idea is based on using two keys DHK created and managed using Diffie-Hellmen algorithm, which is responsible for encrypting SK1 of length N through multi algorithms. SK1 is going to be sent as SMS over GSM for the second party of the communication. More illustration is in the following steps:

4.4.5.1 Preparation for key distribution over GSM

Assume that we have two parties P_A and P_B are going to communicate over LAN in a secure channel.

Assume the P_B is connected to P_A at time $T_{connect}$.: Time stamp will be encrypted sequentially depending on the modulus of $T_{connect}$ over 5, The result will be the index of the algorithm which responsible for encryption the T_{send} later, then each timestamp will encrypt decrypt by incrementing 1 to current selected algorithm.

1. Diffie-Hellman Algorithm

    1.1 P_A and P_B are choose a random private numbers (a) and (b).

    1.2 Share public random (g) and (p).

    1.3 PA send to P_B g^a mod p as P_AS.

1.4 PB send to P_A g^b mod p as P_BS.

1.5 DHK is calculated between parties as following:

a) DHK = $(P_BS)^a$ mod p for P_A

b) DHK = $(P_AS)^b$ mod p for P_B

Example a=6, b=15, p=23, g=5

$P_AS=5^6$ mod 23= 8

$P_BS=5^{15}$ mod 23= 19

DHK on P_A is 19^6 mod 23=2

DHK on P_A is 8^{15} mod 23=2

1.6 Convert DHK to md5 hash to get the SK1 of length N bit to use it as a secrete key for multi-level encryption.

2. Multi-level key encryption

2.1 Generate random key of length N bit SK2. By P_A party.

2.2 Calculate start index of multi-level encryption as $T_{connect}$ mod 5 the value will be in range [0 – 4] which is the number of used predefined algorithm table 4-1 order in both connection parties.

2.3 Now the secret key is SK1 and the data is the second key SK2 is going to pass through the multi-level encryption process.

Table 4-1 Predefined encryption algorithm order in communication parties

Start index	0	1	2	3	4
algorithm	AES	DES	3DES	Rijndael	RC2

2.4 Loop the SK2 through the five algorithms Table 4.1 then return the encrypted key.

2.5 The resulted encrypted key will be sent as SMS to P_B over GSM via Predefined mobile number.

Example: Assume that $T_{connect}$ timestamp= 635888764574724677

Start index= $T_{connect}$%5=2

Then the SK_2 will loop as the following Figure 4.5 and Table 4.1:

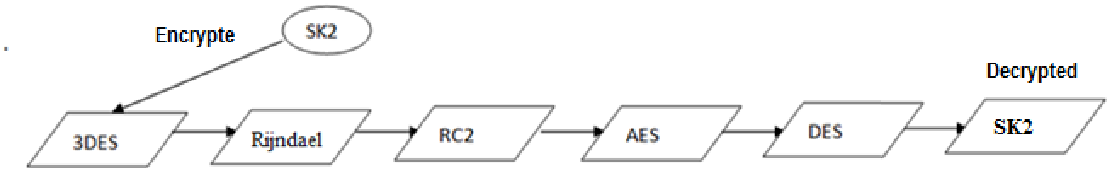


Figure 4-5 levels encrypted key which going to be sent over GSM

4.4.5.2 Receiving key and decryption process

Read the SMS encrypted received key (RK) from the mobile which is already attached to the PC before the communication process initiated.

According to $T_{connect}$ time. Calculate start index of multi-level encryption as $T_{connect}$ mod 5 the value will be ranged in [0 - 4], which is the number of the used predefined algorithms order on both connection parties.

1. Now the secret key is SK1 which explained in 4.3.4.1 and the received key RK are going to pass through in order to reverse the multi-level encryption process to get the clear key SK2.
2. Loop the RK through the five algorithms Table 4.1 then return in reverse order.
3. The result decrypted key (SK2) will be saved in the communication application.

4.4.5.3 Encryption text and transfer

After connection establishment and storing SK2 in the application, communication will be start securely. Each message has T_{send} time stamp which follows the same procedure to encrypt the text by calculating the start index for encryption, in order to minimize the encryption time. The application is going to select a single algorithm and encrypt the text before sending according to the following equation:

Selected Algorithm=T_{send}%5

Selected algorithm is a number in range [0-4], which is the index of predefined algorithms index saved in the application Table 4.1

Example assume T_{send}=635888764574724672

Selected algorithm =635888764574724677%5 = 2

Then the message will be encrypted using SK2 as a secret key and DES as an encryption algorithm.

4.4.5.4 Decryption text and transfer

On the other communication party, detect T_{send} time then follow the same procedure by calculating the selected algorithm, then decrypting the received text.

Example: assume T_{send}=635888764574724672

Selected algorithm =635888764574724677%5 = 2

Then the message will be decrypted using Sk2 as a secret key and DES as a decryption algorithm.

4.5 Conclusion

This chapter discussed the background concepts of the "APK v.1" algorithm [9]; the structural model that will be implemented, the general steps of the algorithm, the key generation algorithm, the encryption algorithm and the decryption algorithm. We also discussed the enhancement on the APK v.1. All the updates that have been made on APK v.1 algorithm will enhance the performance and the strength against the cryptanalysis attack that is what we are going to prove in the next chapter. The performance enhanced by removing sending clear text via SMS, port knocking, number of SMS to be sent. APK v.1 algorithm is time, bandwidth and cost consuming, by reducing knocking, number of sending SMS which reflects on the cost required for connection establishment, one SMS to secure connection, finally encrypt the key which is sent over GSM.

Randomization is the best solution for going through a behavior which is unexpected, the calculated randomization guarantees that each party has its own secrete key, time synchronization, and random selection of algorithm. On the other hand, key length is flexible, and encryption algorithm are flexible to be modified by adding or removing other algorithms, this makes it difficult to the cryptanalysis attacker to find the key, and if the attacker tries to guess the key, the key length makes the job hard and take a long time as mentioned in table 2.2 and if the attacker tries to guess the key, he has to multiply this trials by 2.5 in average of used algorithms.

5 Chapter 5: Experiments and Results

In this chapter, we discuss the experimental design for our research work and the experiments results.

5.1 Experimental Design

This section describes the techniques and simulation choices that were made to evaluate the performance and strength of the KDOGSM v.1. In addition to that, this section will discuss the methodology and the parameters such as: system parameters, experiment factors, and experiment initial settings.

5.1.1 Simulation Setup

All the algorithms of the KDOGSM v.1 system model and simulation have been implemented using C# programming language. We create a graphical user interface (GUI) application to simulate the performance of KDOGSM v.1 and compare with the performance of the APK v.1. We also create a GUI to simulate the strength of APK v.1 and KDOGSM v.1 against the cryptanalysis attacks that tries to get plain text from ciphertext.

This application uses the provider classes in .Net framework to simulate the performance of KDOGSM AES, DES, RC2, 3DES and Rijndael implementation, This provider class is implemented, tested and optimized to give the maximum performance for the algorithm, used encryption/ decryption algorithms. The implementation managed wrappers for AES, DES, 3DES, RC2 and Rijndael available in System.Security.Cryptography.

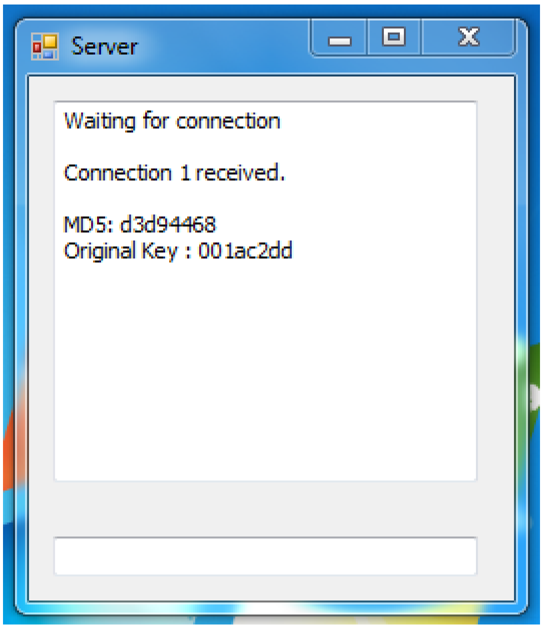
The System.Security.Cryptography namespace provides cryptographic services, including secure encoding and decoding of data, as well as many other operations, such as hashing, random number generation, and message authentication [40].

As mentioned before in APK v.1.The whole process is initiated by user by sending an SMS to the SMS Server connected to the PRNG requesting for the 256-bit One-Time Key which will act as the key for AES encryption and an 8-bit random number R which should be relatively prime to P and N. The numbers P and Q are prime numbers unique for each user and are already stored in the database server connected to the SMS server for each user. The SMS is sent through a dedicated channel which in out of band for general communication.

The reply sent by the server is also send through the same dedicated channel. The only precursor for this process is that the user's mobile number should be registered beforehand with the SMS Server. The message send by the user should be of the form Request OTP <Allotted User ID><Fixed User Pass-code><source IP-Address> The OTP is generated with validity time period so that there is no duplicate OTP generated before the use of original OTP.

The simulation program main form shown below figure 5.1, 5.2 and 5.3 views the graphical user interface which is going to be used to encrypt/decrypt the key and the messages transferred over the LAN between communication parties.

Figure 5.1 is the Graphical user interface for the party how responsible for organizing the communication. This terminal act as server (P_A) which listening over LAN on specific port. The terminal keeps listening until receiving a connection from another party P_B.



Waiting for connection

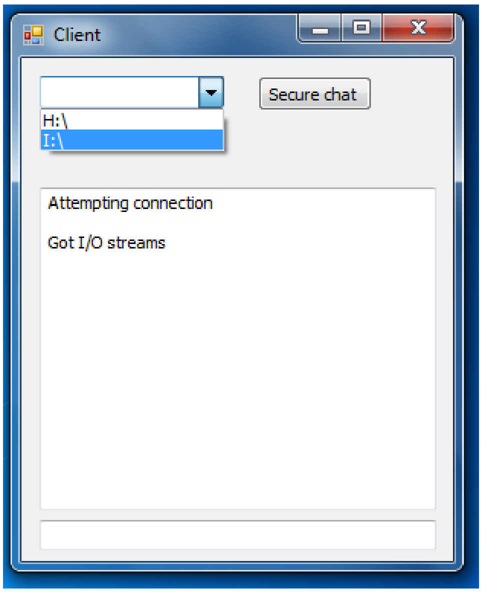
Connection 1 received.

MD5: d3d94468
Original Key : 001ac2dd

Figure 5-1PA UI which responsible for key generation and sending over GSM

Party (P_B) has a predefined mobile number on P_A, after receiving the connection our algorithm starts to generate keys and send it over GSM to the mobile which is already defined before.

Terminal P_B starts working after receiving the SMS via the mobile which is already attached to the terminal, this form works only if there is a mobile phone attached to the terminal. After receiving the key, the form decrypts it according to the time stamp and algorithm order, then the communication becomes secure and text transfer starts safely over the LAN



```
Attempting connection

Got I/O streams
```

Figure 5-2PB UI response for read the SMS from the mobile and decrypt it

After running cryptanalysis that used to guess the key which is able to decrypt the text and get the plain text. The cryptanalysis scenario assumes that the attacker knows the key length for each session to encrypt the text, and he should guess the other synchronization settings of the KDOGSM v.1 algorithm. According to that scenario, we have saved sessions conversation in two ways: encrypted and plain text, session key and the algorithm which is used to encrypt the text.

Cryptanalysis starts to read the log mentioned before and tries to guess the key after we provide the key length for each key the cryptanalysis decrypts the clear text to cipher text and compare the result with the log, then tests the key for the log set.

5.1.2 System Parameters

The experiments conducted use the following:

1. Software: The simulation program is compiled using the default settings in .NET 2012 visual studio for C# windows applications, Windows 7.
2. Hardware: LAPTOP: Intel® Core™i7 2410M CPU N270 @2.30GHz 8GB of RAM. Intel® 82579V Gigabit Network Connection.

The experiments will be performed more than once to assure that the results are consistent and valid to compare with the different algorithms. The simulation program is running inside the LAN environment; see figure 5.3.

Figure 5-3Local Area Network

5.1.3 Experiment Factors

The evaluation of the KDOGSM v.1 algorithms is done by using methods such as:

5.1.3.1 Comparison

Time, Total cost and Bandwidth. Several performance metrics are collected, such as: the encryption time is considered as the time that an encryption algorithm takes to produce a cipher text from a plaintext.

Total cost: the total cost is one of the metrics which make applying of the algorithms possible, the cost of both APK v.1 and KDOGSM v.1 are based on the number and the cost of the SMSs being used during the process.

Bandwidth: bandwidth is considered as one of measure performance matrix of an algorithm, the good use of the bandwidth means high optimization of both algorithm and LAN.

5.1.3.2 Cryptanalysis

We will use the cryptanalysis according to a specific scenario that can be used for this purpose. Because it is hard to find a tool that can be used in general, all the tools created and published for a specific algorithm.

There are many parameters attacker should know before starting his cryptanalysis:

1. Attackers should know the algorithms used in encryption/decryption process.
2. Attackers should know the order of these algorithms in the system.
3. Attackers should know $T_{Connect}$ and T_{send} to control the order of the algorithm.
4. The attacker should know the key length which is used in the encryption/decryption process.

Without knowing all these algorithm parameters, one possible attack under these circumstances is the brute-force approach of trying all possible keys, and guessing which algorithm is used for each guessed Key.

5.1.4 Simulation Procedure

In the experiments, the following tasks that will be performed are shown as follows:

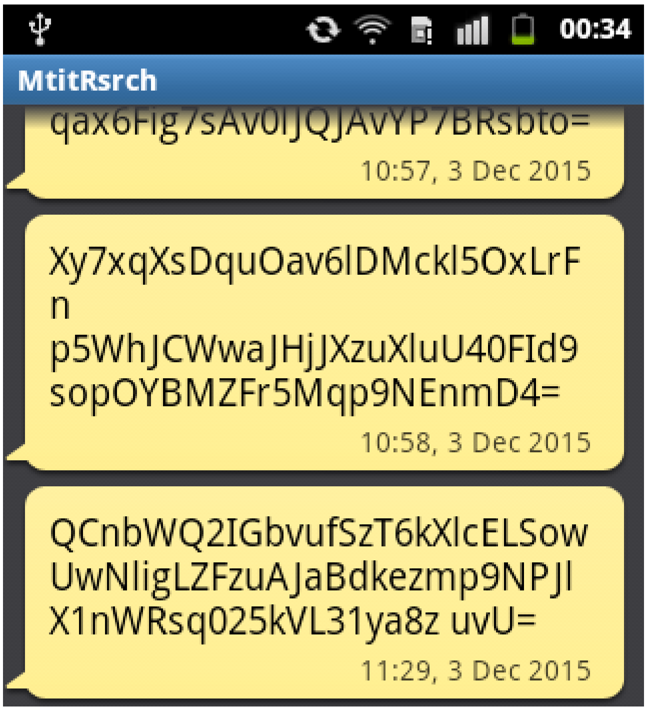
1. Validate key generation changed for each session.
2. Validate algorithms if they changed for each time stamp.
3. Prepare list of encrypted/decrypted session which was saved before from previous communication session.
4. Applying the guessed key with the five algorithms.
5. Calculate required time to decrypt the text with valid key and algorithm mathematically.

5.2 Experimental Results

This section will show the results which are obtained by running the simulation program using different data. The results show the impact of changing key length on each session and the impact of cryptanalysis attack against the KDOGSM v.1.

5.2.1 Validation of key generation for each session

As shown in figure 5.4 the key which is sent over GSM to the party who is going to communicate is changed per each connection and sent to the predefined mobile number in period not more than 20 seconds, if the period exceeds 20 seconds the session is terminated. The SMS content presents the encrypted key results from multi encryption process, which is going to be decrypted by reversing the process on the client application.

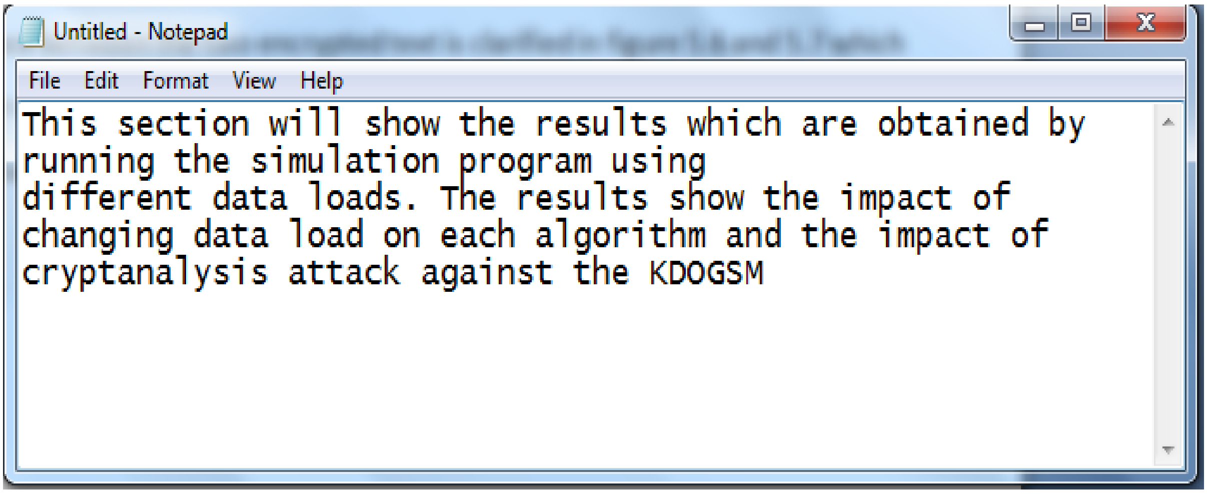


```
qax6Fig7sAv0lJQJAvYP7BRsbto=
10:57, 3 Dec 2015

Xy7xqXsDquOav6lDMckl5OxLrF
n
p5WhJCWwaJHjJXzuXluU40FId9
sopOYBMZFr5Mqp9NEnmD4=
10:58, 3 Dec 2015

QCnbWQ2IGbvufSzT6kXlcELSow
UwNligLZFzuAJaBdkezmp9NPJl
X1nWRsq025kVL31ya8z uvU=
11:29, 3 Dec 2015
```

Figure 5-4Received key over GSM

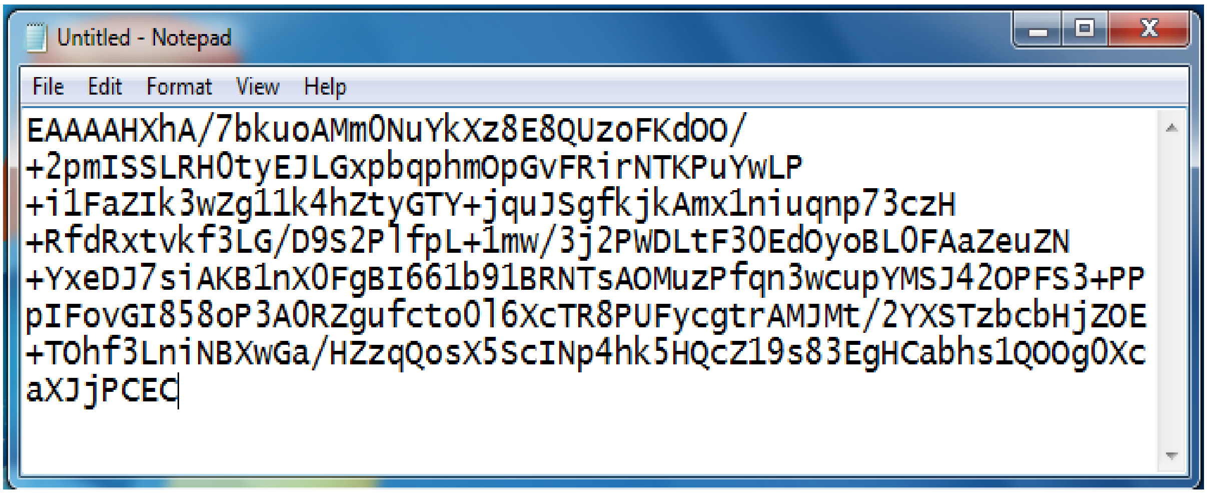
5.2.2 Validation that algorithms are changed for each time stamp

Figure 5.5 shows a plain text. Its encrypted version and the decrypted version using KDOGSM v.1 algorithm are shown in figure 5.6 and 5.7 in same session, 5.8 is the encrypted text in another session for the same plain text. The difference between the two encrypted texts is clarified in figure 5.7 and 5.8 which proves the principle of algorithm changing according to timestamp T_{send}. The encryption and decryption processes are applied between two computers using LAN.



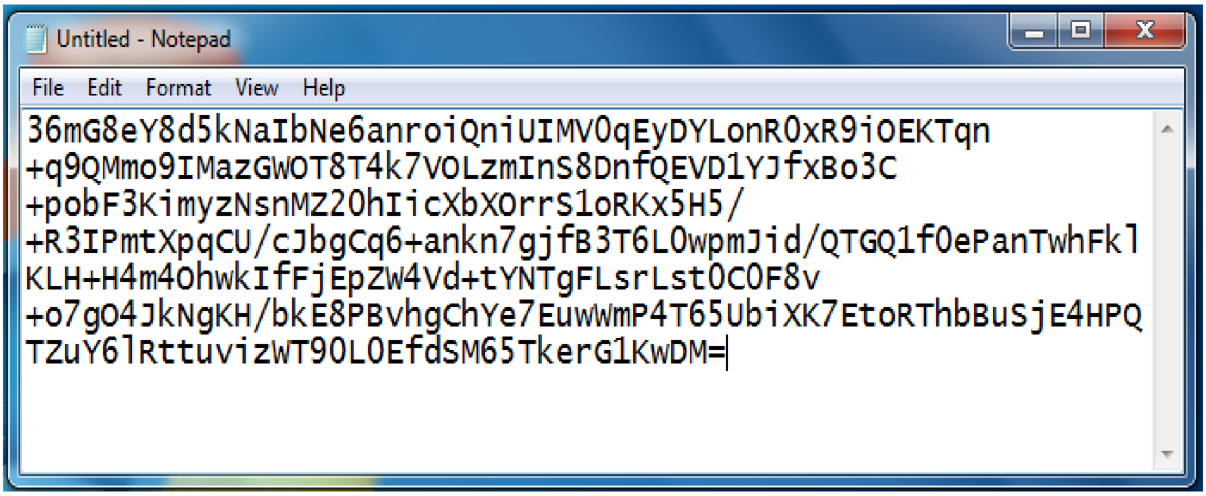
```
This section will show the results which are obtained by
running the simulation program using
different data loads. The results show the impact of
changing data load on each algorithm and the impact of
cryptanalysis attack against the KDOGSM
```

Figure 5-5Plain text



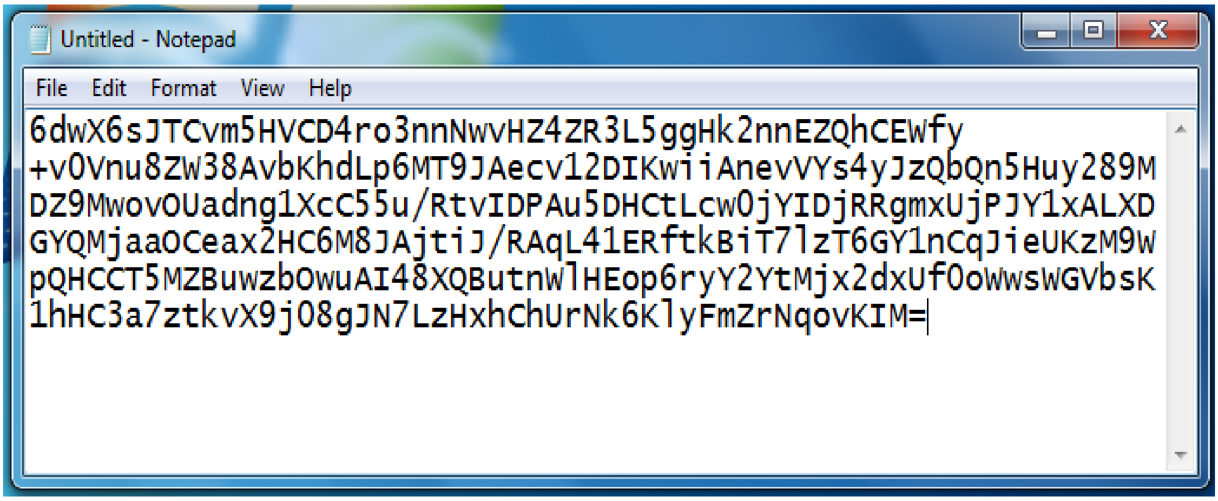
```
EAAAAHXhA/7bkuoAMm0NuYkXz8E8QUzoFKdOO/
+2pmISSLRH0tyEJLGxpbqphm0pGvFRirNTKPuYwLP
+i1FaZIk3wZg11k4hZtyGTY+jquJSgfkjkAmx1niuqnp73czH
+RfdRxtvkf3LG/D9S2PlfpL+1mw/3j2PWDLtF30Ed0yoBL0FAaZeuZN
+YxeDJ7siAKB1nX0FgBI661b91BRNTsAOMuzPfqn3wcupYMSJ42OPFS3+PP
pIFovGI858oP3A0RZgufcto0l6XcTR8PUFycgtrAMJMt/2YXSTzbcbHjZOE
+T0hf3LniNBXwGa/HZzqQosX5ScINp4hk5HQcZ19s83EgHCabhs1QOOg0Xc
aXJjPCEC
```

Figure 5-6the encryption of the plaintext during session using one of the algorithm



```
36mG8eY8d5kNaIbNe6anroiQniUIMV0qEyDYLonR0xR9iOEKTqn
+q9QMmo9IMazGWOT8T4k7VOLzmInS8DnfQEVD1YJfxBo3C
+pobF3KimyzNsnMZ20hIicXbXOrrS1oRKx5H5/
+R3IPmtXpqCU/cJbgCq6+ankn7gjfB3T6L0wpmJid/QTGQ1f0ePanTwhFkl
KLH+H4m4OhwkIfFjEpZW4Vd+tYNTgFLsrLst0C0F8v
+o7gO4JkNgKH/bkE8PBvhgChYe7EuwWmP4T65UbiXK7EtoRThbBuSjE4HPQ
TZuY6lRttuvizWT90L0EfdSM65TkerG1KwDM=
```

Figure 5-7the encryption of the plaintext during session using another of the algorithm



```
6dwX6sJTCvm5HVCD4ro3nnNwvHZ4ZR3L5ggHk2nnEZQhCEWfy
+v0Vnu8ZW38AvbKhdLp6MT9JAecv12DIKwiiAnevVYs4yJzQbQn5Huy289M
DZ9MwovOUadng1XcC55u/RtvIDPAu5DHCtLcw0jYIDjRRgmxUjPJY1xALXD
GYQMjaaOCeax2HC6M8JAjtiJ/RAqL41ERftkBiT7lzT6GY1nCqJieUKzM9W
pQHCCT5MZBuwzbOwuAI48XQButnWlHEop6ryY2YtMjx2dxUf0oWwsWGVbsK
1hHC3a7ztkvX9j08gJN7LzHxhChUrNk6KlyFmZrNqovKIM=
```

Figure 5-8 same plain text with same algorithm but in another session

5.2.3 Comparing processing results of APK v.1 and our KDOGSM v.1:

1. Connection initialization:

    APK v.1: connection always starts by sending an SMS.

    KDOGSM v.1: Just connect over LAN to the other communication party.

2. Establishing the connection:

    APK v.1: SMS received with plain text to predefined mobile number.

    KDOGSM v.1: SMS received with encrypted text to predefined number

3. Bandwidth optimization:

    APK v.1: Port knocking by sending Synchronization packets to validate the pattern to open the port.

    KDOGSM v.1: Connection established after decrypting the SMS with no knocking

4. Cost minimization:

    APK v.1: cost price of 2 SMS Over special channels

    KDOGSM v.1: Cost on SMS over GSM channels

5. Complexity of the algorithm:

    APK v.1: Use single encryption/decryption algorithm AES.

KDOGSM v.1: Use an array of encryption/decryption algorithms, choosing an algorithm from the array randomly according to T_{send} time. According to Table 2.2 there is a list of Time estimation for guessing the alternatives of each key length. Such table is valid if we got one encryption/decryption algorithm. But in our case we got single algorithm from random five algorithm. So for each trial of guessing a key, the attacker has to pass through the five algorithms to determine whether the key is valid for the session or not, then we can multiply the required time * 2.5 for the average of picking the right algorithm among the five algorithms.

Table 5.1 Sample of key guessed by the cryptanalysis

Sess. id	GUESSED KEY	STATUS_CD	START_DT	END_DT
1	6cad7c2f	FAIL	54:00.0	54:00.0
1	68061f05	FAIL	54:00.0	54:00.0
1	30ef6464	FAIL	54:00.0	54:00.0
1	bffb4ddf	FAIL	02:00.0	02:00.0
1	aaa0cca8	FAIL	02:00.0	02:00.0
1	17cda1ca	FAIL	02:00.0	02:00.0
1	c89e057e	FAIL	02:00.0	02:00.0
1	15bbfc6a	FAIL	02:00.0	02:00.0
1	347cae7e	FAIL	02:00.0	02:00.0
1	2beefc3f	FAIL	02:00.0	02:00.0
1	52b7c8e4	FAIL	02:00.0	02:00.0
1	4ad3cbd3	FAIL	02:00.0	02:00.0
1	a2947147	FAIL	02:00.0	02:00.0
1	358464fa	FAIL	02:00.0	02:00.0
1	d5d88fc3	FAIL	02:00.0	02:00.0
1	33f2d22f	FAIL	02:00.0	02:00.0
1	a053335d	FAIL	02:00.0	02:00.0
1	710ca1f9	FAIL	02:00.0	02:00.0
1	999a2e50	FAIL	02:00.0	02:00.0
1	cd767b18	FAIL	02:00.0	02:00.0
1	744c7c9c	FAIL	02:00.0	02:00.0
1	7c962eb5	FAIL	02:00.0	02:00.0
1	188558d4	FAIL	02:00.0	02:00.0
1	62e5c42a	FAIL	02:00.0	02:00.0
1	62b7b199	FAIL	02:00.0	02:00.0
1	83bb25f4	FAIL	02:00.0	02:00.0
1	850fd083	FAIL	02:00.0	02:00.0
1	232ff89f	FAIL	02:00.0	02:00.0
1	41495ce5	FAIL	02:00.0	02:00.0
1	1d3cd325	FAIL	02:00.0	02:00.0
1	b4773204	FAIL	02:00.0	02:00.0
1	33894bb7	FAIL	02:00.0	02:00.0
1	cbfb0487	FAIL	02:00.0	02:00.0
1	fb22f818	FAIL	02:00.0	02:00.0
1	53fbb01a	FAIL	02:00.0	02:00.0
1	735703ff	FAIL	02:00.0	02:00.0
1	c4310fc5	FAIL	02:00.0	02:00.0
1	d403f248	FAIL	02:00.0	02:00.0
1	e09d8c44	FAIL	02:00.0	02:00.0
1	59e18485	FAIL	02:00.0	02:00.0
1	52d464b6	FAIL	02:00.0	02:00.0
1	32a8c8f5	FAIL	02:00.0	02:00.0
1	89d3bcd9	FAIL	02:00.0	02:00.0

According to simple calculation:

1. Assume that to find the correct algorithm the average of the used algorithms 2.5.
2. Assume the key combination of alpha and numbers 26*2+10=62 probability.
3. Comparing APK with KDOGSM The following chart.

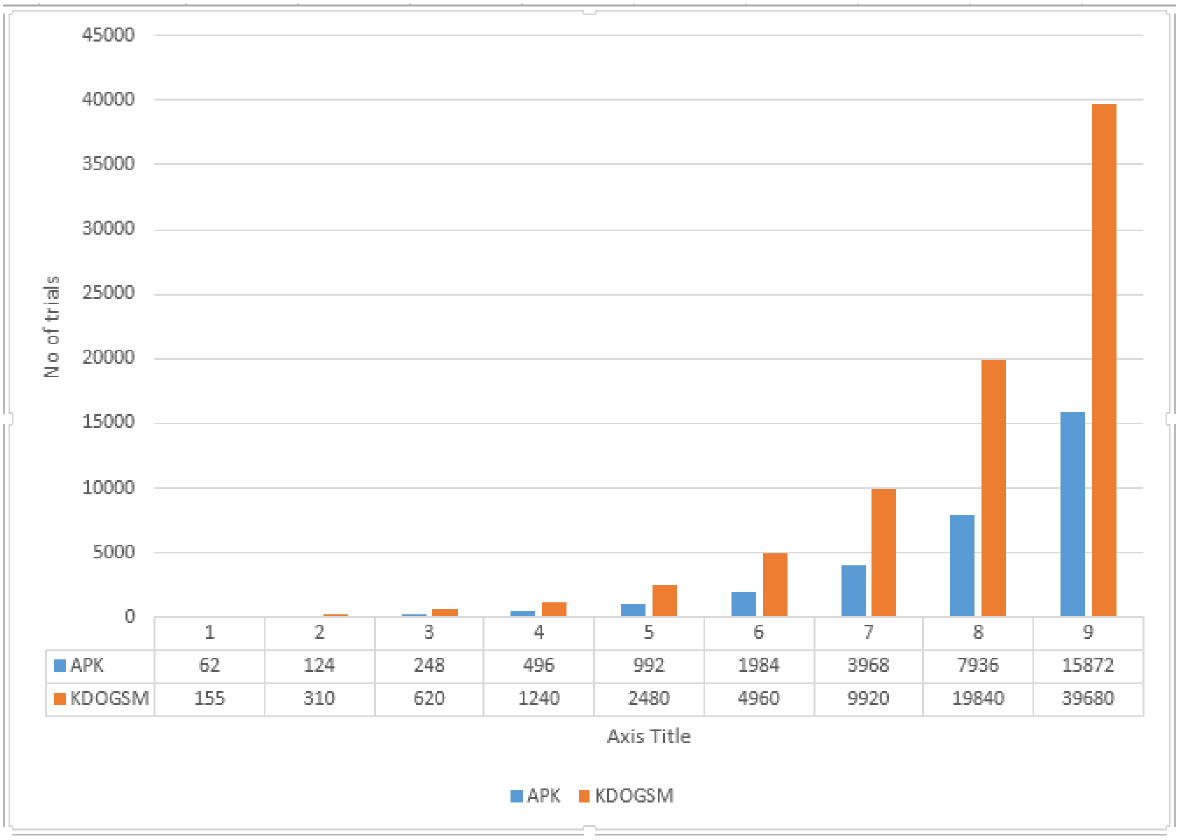


Figure 5-9Number of guessing key to decrypt the content

From the previous sample figure 5.9 and by calculating the numbers of trials to guess the valid key as we provide before the guessing process.

Number of tries for

First session was 9681085 trials in 337 minute,

Second session 36566119 trials in 533 minute.

These numbers are just for guessing the given key. We can multiply these numbers in 2.5 as average algorithms. The resulted numbers are the real trials for the decryption using cryptanalysis method.

5.2.4 Time Estimation for generating Key of Length N:

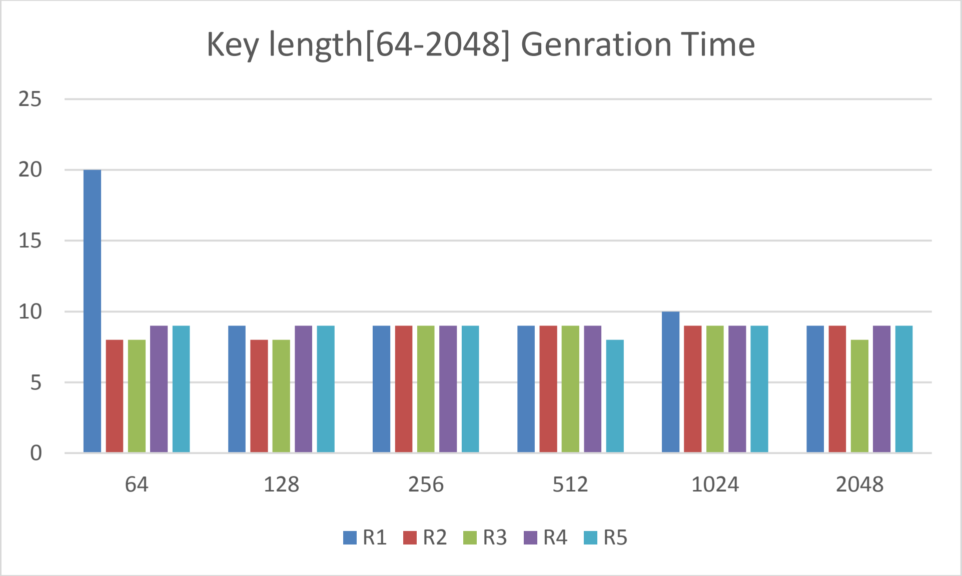


Figure 5-10Time estimation in ms for each key length

As shown in Figure 5.10 it is clear that the time estimation for generate a key of length 64, 128, 256, 512, 1024, 2048 seems to be equals in the time but on anther hand we have to take in our calculation the SMS Cost for each as shown in Table 5.2

Table 5.2 SMS cost for each Key Length

Key length	SMS COST
64	1 SMS
128	1 SMS
256	1 SMS
512	1 SMS
1024	2 SMS
2048	3 SMS

So if we are going to select the optimal key length which responsible for maximizing the performance, strength, and less cost is key of 512 length.

5.3 Conclusion

This chapter described the techniques and simulation choices made to evaluate the performance and strength of the APK and KDOGSM algorithms. In addition to that, this chapter discussed the methodology related parameters such as: system parameters, experiment factors, and experiment initial settings.

In the performance evaluation results of algorithm KDOGSM is better than APK algorithm in multi criteria as mentioned before, the main criteria is the key transfer between two clients, either the financial cost of connection initialized is decreased 50% than APK algorithms, the bandwidth of the network is optimized by remove the port knocking, and establish the connection from first login, in our algorithm we enlarge the guessing dataset of the brute force attach by multiplying the initial dataset in 2.5 as number of trials to guess the key and algorithms which used to encrypt the packet which sniffed from the LAN.

6 Chapter 6. Conclusions and Future Directions

In this chapter, we present and discuss the conclusions for this work and the future directions

6.1 The conclusion

The Cryptography provides security to the data that is transferred. The security of cryptographic system relies on the fact that the cryptographic keys are secret and known only to a legitimate user. Hence, we enhanced APK version 1 algorithm based on port knocking, clear text key exchange over GSM based on IP address and one-time password key generation which is generated on PRNG [9] so the process of key generation is based on sessions. This creates more complexity to crack or guess the keys by using cryptanalysis techniques; this approach is called the key-updating method which is a new approach to increase the difficulty to discover the key. To break this algorithm, we need to know the IP address of the communicating machine and the random numbers. This thesis presents a performance evaluation of enhanced algorithm APK v.1 and selected symmetric encryption algorithms for KDOGSM version 1 which are AES, Rijindael, DES, 3DES and RC2, as this thesis presents evaluation of the strength of KDOGSM version1 and APK version1 algorithms against cryptanalysis attack to decrypt the cipher text. Several points can be concluded from the experimental results.

1. The KDOGSM, APK Algorithms support the key-updating approach.
2. In case of changing sending time, it was concluded that KDOGSM version1 has a better performance (More complexity for cryptanalysis) than the other symmetric encryption algorithms.
3. APK version1 requires SMS cost more than KDOGSM version1.
4. In case of attacking KDOGSM algorithms using the cryptanalysis attack, where we assume that the attacker knows the key length. It was concluded that the KDOGSM version1 is stronger than the APK version1 against the attack since the randomization of selecting encryption algorithm, but the KDOGSM version1 can be broken in long time in months as average that depends on the speed of the attacker machine and algorithm synchronization setting.
5. The enhanced KDOGSM version1 algorithm process has an advantage, that the key generation is based on session, so in every session, we have a different key, as the key length varies according to configuration of algorithm. The key length can have no boundaries. The key length can be more than 512 bits.

From all the previous concluded results in this thesis, we summarize the most important differences between the APK version1 and KDOGSM version 1 in

Table 6-1 important differences between the APK v.1 and KDOGSM v.1

	KDOGSM version 1	APK version 1
Generated key	Key varies every session Key length varies by settings Key length can be more 512 bits	Varies every session Constant 256 bit
Brute-force attack	Impractical	Impractical
Cryptanalysis	Needs longer Time (months as average)	Succeeds in short time (minutes as average)
Bandwidth	Connects from the first time	Port knocking many request
Cost To establish the connection	One SMS price	Many SMS price

Finally, through our work in this thesis, it can be concluded that we can encrypt and decrypt the user and / or application data very easily and simply we consider it as the spirit of this algorithm.

6.2 Future Directions

Future scope for KDOGSM version1 algorithm can be one of the following directions:

1. Calculate the key length according to size of transfer data over LAN.
2. Enlarge the algorithm to work over WAN, the current solution valid for WAN, but the main issue here is the validity time of the received SMS, the communication parties although concern with the time of receiving the SMS to be as fast as could. In our solution we have restrict the validity time as 20 second at most, if the SMS wont received the communication have to be reinitialized.
3. Generate especial key for each algorithm, our solution use on key for the multi-level encryption process, but in the next edition we are going to set keys for each used algorithm which going to make the brute force attack impractical.

7 References

1. M. B. Krishna, M. N. Doja,” Symmetric Key Management and Distribution Techniques in Wireless Ad hoc Networks”, International Conference on Computational Intelligence and Communication Systems, 2011, PP: 727-739.
2. S. Barman, S. Chattopadhyay, D. Smanta,”An Approach to Cryptographic Key Distribution Through Fingerprint Based Key Distribution Center”, Advances in Computing, Communications and Informatics, 2014, pp: 1629 – 1635.
3. K. K. Khaing, K. Mi Mi Aung, “Secured Key Distribution Scheme for Cryptographic Key Management System”, International Conference on Availability, Reliability and Security,2010, PP: 481-486.
4. M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, V. Shmatikov, ‘The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software’, CCS '12 Proceedings of the 2012 ACM conference on Computer and communications security, 2011, PP: 38-49.
5. The Heartbleed Bug, http://heartbleed.com/, Page updated Apr. 29, 2014[Dec. 7, 2014].
6. Android Security Vulnerabilities, http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html, Jun. 2, 2014 [Dec. 7, 2014]
7. Sur, C.,’A Block Cryptographic Method for Communication Network – its Prospects and Limitations’, Computer and Automation Engineering (ICCAE), 2010 The 2nd International, 2010, PP: 532 – 534.
8. F. Yue; G. Wang; Q. Liu,’A Secure Self-Destructing Scheme for Electronic Data’, Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference, 2010, PP: 651 – 658.
9. Srivastava, V. ; Keshri, A.K. ; Roy, A.D. ; Chaurasiya, V.K. ; Gupta, R. ,’Advanced Port Knocking Authentication Scheme with QRC using AES’, Emerging Trends in Networks and Computer Communications (ETNCC), 2011,PP: 159- 163.
10. One-time password , http://en.wikipedia.org/wiki/One-time_password, Dec. 26, 2014, [Dec.29, 2014].
11. Diffie–Hellman key exchange, http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange, Dec. 29, 2014, [Dec.29, 2014].

12. Y. Yu, J. J urjens, J. Mylopoulos, ‘Traceability for the Maintenance of Secure Software’, Software Maintenance, 2008. ICSM 2008. IEEE International, 2008, PP:297 – 306.

13. N. Smart, ‘ECRYPT II Yearly Report on Algorithms and Keysizes’, Katholieke Universities Leuven, Sep 30, 2012, [Dec. 7, 2014].

14. G. Anderson, R. Michael Varney, P. D. Warren, J. M. Czerwinski, E. G. Andolina,’ Managing Third-Party Relationship Risk’, © 2014 Crowe Horwath International, 2011.

15. Third-Party Risk, federal deposit Insurance Corporation, 2012. [Dec. 7, 2014].

16. M. B. Krishna, M. N. Doja,” Symmetric Key Management and Distribution Techniques in Wireless Ad hoc Networks”, International Conference on Computational Intelligence and Communication Systems, 2011, pp:727-739.

17. W. Stallings, "Cryptography and Network Security - Principles and Practices", Pearson Education Third Edition ed., 2002.

18. Wikipedia (Key exchange), http://en.wikipedia.org/wiki/Key_exchange, Dec. 1, 2013. [Dec. 8, 2015].

19. P. Kandepet, "AN OVERVIEW OF PUBLIC KEY INFRASTRUCTURE," Fall 2013.

20. Symmetric-key_algorithm, Http://en.wikipedia.org/wiki/Symmetric-key_algorithm, Oct. 8, 2014. [Oct. 8,2014].

21. Public-key_cryptography, Http://en.wikipedia.org/wiki/Public-key_cryptography Jul. 15, 2015. [Jul. 15, 2015].

22. J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis," International Journal of Emerging Technology and Advanced Engineering, vol. 1, no. 2, December 2011.

23. B. Schneier, “Applied Cryptography - Second Edition”, John Wiley & Sons, 1996, pp. 210-211.

24. Arjen K. Lenstra and Eric R. Verheul, "Selecting Cryptographic Key Sizes,"J.Cryptology 14(4), pp. 255-293, 2001.

25. Cryptography Engineering: Design Principles and Practical Applications. Ferguson, N., Schneier, B. and Kohno, T. Indianapolis: Wiley Publishing, Inc. 2010. pp. 63, 64. ISBN 978-0-470-47424-2

26. ISO JTC 1/SC 27 (2006). "ISO/IEC 10116:2006 - Information technology -- Security techniques -- Modes of operation for an n-bit block cipher". ISO Standards catalogue.

27. "Search Security", http://searchsecurity.techtarget.com/definition/cryptanalysis, Jun. 2, 2015. [Jun. 2, 2015].

28. Priyanka.M, Lalitha Kumari.R, Lizyflorance.C and John Singh. K, "A New Randomized Cryptographic Key Generation Using Image," in International Journal of Engineering Science and Innovative Technology (IJESIT), Volume 2, Issue 6, November 2013

29. Abd Elminaam, Diaa Salama; Abdual Kader, Hatem Mohamed; Hadhoud, Mohiy Mohamed, "Evaluating The Performance of Symmetric Encryption Algorithms," in International Journal of Network Security, Vol.10, No.3, PP.213–219, May 2010.

30. OP Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi, "Peformance Analysis Of Data Encryption Algorithms," IEEE, 2011.

31. Mandal, Bijoy Kumar ; Bhattacharyya, Debnath; Bandyopadhyay and Samir Kumar, "Designing and Performance Analysis of a Proposed Symmetric Cryptography Algorithm," in 2013 International Conference on Communication Systems and Network Technologies, IEEE, 2013.

32. “Performance Comparison: Security Design Choices”, http://msdn.microsoft.com/en-us/library/ms978415.aspx, Jan. 1, 2016. [Jan. 1, 2016].

33. “Channel (digital image)”, http://en.wikipedia.org/wiki/Channel_ (digital_image), Mar. 15, 2015. [Mar. 15, 2015].

34. "ChilKat Software", http://www.chilkatsoft.com/encryption-features.asp, Jun. 15, 2014. [Jun. 15, 2014].

35. A. A. Tamimi, "Performance Analysis of Data Encryption Algorithms", Retrieved Oct. 1, 2008 [Jun. 15, 2014].

36. Mansoor Ebrahim; Shujaat Khan; Umer Bin Khalid; “Symmetric Algorithm Survey: A Comparative Analysis” in 2013 International Journal of Computer Applications, 2013; pp 12-19.

37. Hansche, "Cryptography", (ISC) 2 Press, 2003.

38. Fiskiran, A.M.; Lee, R.B. Performance Impact of Addressing Modes on Encryption Algorithms, Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Conference on 23-26 Sept. 2001 Page(s):542 – 545.

39. Najib A. Kofahil, Turki Al-Somani2 and Khalid AiZamil3, Performance Evalutionof Three Encryption/Decryption Algorithms.

40. "Microsoft developer network - System.Security.Cryptography", https://msdn.microsoft.com/en-us/library/system.security.cryptography(v=vs.110) .aspx, Mar. 15, 2016. [Mar. 15, 2016].

41. “Information security”, https://en.wikipedia.org/wiki/Information_security, Mar. 2, 2016. [Mar. 2, 2016].

42. “Virtual_private_network”, https://en.wikipedia.org/wiki/Virtual_private_network Mar. 10, 2016. [Mar. 10, 2016].

43. “One-time_password”, https://en.wikipedia.org/wiki/One-time_password, Mar. 15, 2016. [Mar. 15, 2016].

44. Ritika kajal; Deepshikha Saini; Kusum Grewal; “Virtual Private Network”, International Journal of Advanced Research in Computer Science and Software Engineering, 2012; pp 428 - 432.

45. Amna S.M Abukeshipa; Tawfiq S.M Barhoom, “Implementation and Comparison of OTP Techniques (TOTP,HOTP,CROTP) to Prevent Replay Attack in RADIUS Protocol”,Scientific journal of Palestine Applied polytechnic, 2014.