DESIGN AND IMPLEMENTATION OF SECURITY SOLUTION FOR M-GOVERNMENT ON MOBILE PLATFORMS USING HYBRID OF NTRU-PKI AND AES-RIJNDAEL

[1]MALIK ANAS TAWFEEQ, [2]MOHAMAD T. SULTAN

[1]College of Computer Science & Information Technology, Universiti Tenaga Nasional, Putrajaya, Malaysia

[2]Departemt of Computer Science, Cihan University-Erbil, Iraq

E-mail: [1]malikwatari@gmail.com, [2]mohamadtaha@cihanuniversity.edu.iq

ABSTRACT

In recent years and specifically in the era of rapid technology development, many changes have taken place in the field of communication technologies (ICT), where mobile devices have replaced computers in multiple significant tasks. This has influenced the interactions between citizens and government agencies in what is called the m-Government, which is an extension of the e-Government that provides services to citizens in general or subscribers in particular. As these services range from public to private bodies and data transmitted sometimes require authentication and confidentiality, the need to secure them has been found to be inevitable. The problem faced here is regarding security of transmitted data as cryptography algorithms whether symmetric or asymmetric; suffer from issues of key exchange mechanism or security performance. In this research, the authors proposes a system technique to secure the transmission of the m-Government using hybrid security algorithms of AES-Rijndael and NTRU, with concentration on confidentiality and authentication as core services in m-Government current transactions, in order to enhance the security of transmission medium and achieve better security of m-Government services. The findings of this dissertation have shown proof of the powerful presence of security factors concerning confidentiality and authentication in the field of m-Government, and test performance approves that the proposed technique is applicable on smartphone devices.

Keywords: *AES; NTRU; M-Government; Confidentiality; Authentication;*

1. INTRODUCTION

Establishing communication, or becoming the “middle-medium’ between different government agencies and citizens has become indispensable in this century, especially if we were to compare it with the last century before the inventions of mobile devices and wireless network. This is due to the rapid development of technology and communication infrastructure in terms of business, communication, information, and technology. The m-Government (Mobile Government) can be considered as an opportunity in the transformational government strategy. SMS (Short Message Service) can act as one of the avenues through which the m-Government transactions can be carried out. Dealing with such topic means that there are many dimensions involved, and it is getting complicated day by day due to the increasing number of big issues that have to be considered such as privacy, authentication and confidentiality. Important demands of these services must be considered in order to meet user requirements. To achieve this, many security algorithms have been implemented to satisfy requirements dictated in specific security systems. Some have been used as powerful tools for encryption, and some have been broken or even anticipated to be broken soon [1].

Authentication service is a very important process in m-Government services. For that reason many implementations have been done in this field to adopt and enhance this feature. For example, in Sweden, the medical care section of the Uppsala County Council enables citizens to get web-based access to their personal medical records. Access to this privacy sensitive data requires a secure user authentication. The applied authentication scheme is similar to the one frequently used in the course of m-Banking and makes use of SMS technology. In order to get access to the personal medical records, users have to register first. During the registration process, the user's mobile phone number is stored in the system together with newly created username/password based account data. After completion of the registration process, the user can be authenticated reliably. To do so, the user first enters username and password. If these credentials can be verified successfully by the system, a SMS containing a one-time password TAN (Tax Deduction Number) is sent to the user's mobile phone. Finally, the user has to enter the received TAN code in order to successfully complete the authentication process [2].

Considering mobile devices in general, different vulnerabilities have been identified. First, the untrustworthy interface has been mentioned as serious vulnerability. As modern mobile devices have already become almost as complex as PCs (Personal Computer), these devices are nowadays vulnerable to malware such as viruses and hacking. Hence, the trustworthiness of interfaces to mobile devices and high privacy transactions cannot be taken for granted any longer. For that reason, adopting and attaching confidentiality feature to this research is a core concept to enhance security in m-Government [3].

2. MOBILE GOVERNMENT (M-GOVERNMENT)

In order to avoid mixing between e-Government and m-Government, it is important to show the distinction between these two idioms. E-Government refers to the government's use of information technology to send and receive information and provide services to citizens, in other words e-Government could refer to the use of wired internet technology practiced by public organizations to provide better services in efficient manner [4].

Mobile technology is significantly expanding the governments' capacity to produce benefits and deliver outcomes for governments, citizens, businesses, and to impact positively national overall economic growth. The most notable progress will be in developing countries which have been historically restricted by poor or non-existent communication infrastructure that, in turn, has also stunted their economic development and social improvements. However, m-Government development will also provide countries with more developed e-Governments and the opportunity to tackle a number of issues - such as those related to the digital-divide - which remain a critical factor in the levels of e-government services take-up which are lower than expected in many countries [4]. M-Government can also be seen as strategic enforcement which will employ all sorts of wireless and mobile technologies, applications and devices towards enhancing the quality of service delivery to all e-Government key players including citizens, business organizations and a variety of government departments [5].

The m-Government services are laden with ubiquity as its most prominent strength, a concept used to describe the provision of information and services at whichever place and time, upholding the idea of personalization, ease of use, time and cost saving and services which are based on various locations. Many countries are currently strong advocators of m-Government services, such as the USA, the UK, Singapore, Malaysia and Australia. What stands out to be an important principal is that in an e-Government transaction, the involved parties are all authenticated securely, and any transmission of information will be treated with confidentiality and integrity. These security requirements have been made more significant and are more emphasized with the emergence of the m-Government, due to the fact that the wireless interfaces have been proven to have some verified security deficiency if they are to be drawn in comparison with their wired counterparts. Additionally, the ever-increasing storage and processing capabilities of mobile devices have seized the attention of malevolent programmers and hackers all over the world [6].

To look at it as a whole, four major models of m-Government have emerged: government-to-citizens (G2C), government-to-government (G2G), government-to-business (G2B), and government-to-employees (G2E). Mobile applications and services largely constitute Government-to-Citizens (G2C) services. Nonetheless, G2G, G2B and G2E m-government services are also established. In this research the concentration is placed on Government-to-Citizens (G2C) services as a core approach, and more specifically on securing EMR (Electronic Medical Records). Whether or not they are interactive (e.g. alert messages), educational (e.g. grades, admissions, exam results), or transactional (e.g. bank account info), such services must be secured against different types of attacks and breaching [4].

3. RELATED WORKS

This section lists some of literature work accomplished in the field of m-Government; the researchers have analyzed this literature based on security strengths and weaknesses.

The authors in [28] have made an early adopting of m-Government business to employee in the year of 2001. The aim of their work was to promote using of m-Government in communication between agency and employees but no secure medium is provided.

The authors in [26] have proposed an architecture for implementing the Mobile Government Services in Korea. Which is a PDA based interactive services m-police and m-tax management. The weakness of this architecture is that the services were based on platforms of private service providers.

The authors in [27] have proposed a technique of SMS Texting to Encourage Democratic Participation by Youth Citizens where they have suggested adopting M-voting to be used in elections. However this technique Suffered lack of security and privacy.

Sameer Hasan Al-bakri [25] has proposed a novel peer-to-peer SMS security solution which provided confidentiality and compatible SMS transmission but suffered from weak authentication.

The above mentioned papers discussed different techniques tried to decrease the gap between mobile wireless services and government agencies. However these implementations lack to one or more security issues which needs to be treated.

4. ELECTRONIC MEDICAL RECORDS (EMR)

An EMR is usually a computerized legal medical record created in an organization that delivers care, such as hospital and doctors' surgery [7]. EMR tends to be a part of a local stand-alone health information system that allows storage, retrieval and manipulation of records and reduces medication errors. Data and information of EMR include patients' demographics, progress notes, problems, medication, vital signs, past medical history, immunizations, laboratory data, and radiology reports. The EMR has the ability to generate a complete record of a clinical patient's encounter, as well as supporting other care-related activities directly or indirectly through interface including evidence-based decision support, quality management, and outcomes reporting. An electronic record may be created for each service to a patient, such as radiology, laboratory, or pharmacy, or as a result of an administrative action (for example, creating a claim). Some clinical systems also allow electronic capture of physiological signals (for example, electrocardiography), nursing notes, physician orders, etc. More attention is now paid for the security, privacy and confidentiality as the challenge for the developers and providers of e-Health is to set and meet security standards of delivery that ensure consumers are able to use these services safely and with confidence. Articles have discussed issues related to Privacy, Confidentiality and Security in the Journal of Medical Internet Research from 2006 till 2011 and these are shown in Figure 2.1 shows that the articles on security, confidentiality and privacy are on increase. In 2006, only 9 articles have been published but in 2010 they have increased to 25 [8] [9].

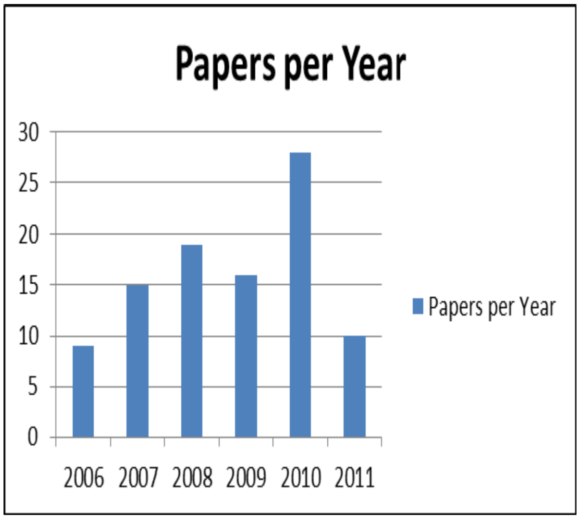


Figure 1: Articles discussed privacy, confidentiality and security in the journal of medical internet research from 2006 till 2012 [10]

5. CRYPTOGRAPHY

Cryptography is the science of keeping messages secure and it is practiced by cryptographers [29]. Another word related to this field is cryptology which combines the cryptography and cryptanalysis and mathematical processing, and the people working with it are known as cryptologists. Encryption on its own does not provide security, the encryption and decryption operations must be governed by a proper process. There is no accurate data on the cost of failures in the security of the information infrastructure, and the reason for this is that often, the victims rarely make security compromises public, for fear of embarrassment, or concern of being incurred the cost of punitive damages for inadequate protection of private information or loss of business [11].

In this research, cryptography is most concerned with security algorithms and their build system in terms of encryption and decryption performance. In order to choose appropriate algorithms to secure the transfer medium in the m-Government, a comparative study among different cryptographic algorithms (symmetric and asymmetric) must be carried out. Different algorithms provide different levels of security, and it depends on how robust they are to break. There are different criteria to determine the risk of breaking cryptographic algorithms, for example, if the time required to break an algorithm is longer than time needed to keep the encrypted data secret, then the algorithm is seen to have achieved the goal of security. Also if the cost required to break a specific algorithm is larger than the value of the data ciphered, then it is considered a safe algorithm. The same would apply for the amount of data that are encrypted with a single key [11] [12]. The following sections shall be shedding light on some of the well-known algorithms applied in the field of security.

5.1 Symmetric Cryptography

Before we define the symmetric cryptography algorithm, first of all, we need to define the secret key, which is a single key that is used to encrypt and decrypt texts. This process is also known as secret-key cryptography, Symmetric algorithms [31], which can also be labeled as the conventional algorithms, are algorithms in which their encryption key can be computed from the decryption key, and works the opposite way as well. In a lot of symmetric algorithms, the encryption key and the decryption key do not show any difference. These algorithms, by other names called the secret-key algorithms, single key algorithms, or one-key algorithms, are pre-conditioned that the sender and receiver would come to a mutual decision on a key before communication can take place safely. The protection offered by a symmetric algorithm is vested within the key; where exposing the key would imply that just about anyone is at liberty to encrypt and decrypt countless number of messages. Provided that the communication should stay discreet, it is imperative that the key must also remain as such [12].

Symmetric cryptography suffers from some drawbacks. For example, the process of exchanging the secret key requires high level of trust as the process of choosing, delivering and storing the secret key is not easy to be done in a secure and dependent manner. Symmetric key encryption also lacks the authentication service; in other words the recipient can neither authenticate the sender nor verify that the decrypted message is the same as the original message [13]. Examples of well-known symmetric algorithms include DES, 3DES and AES [13][14].

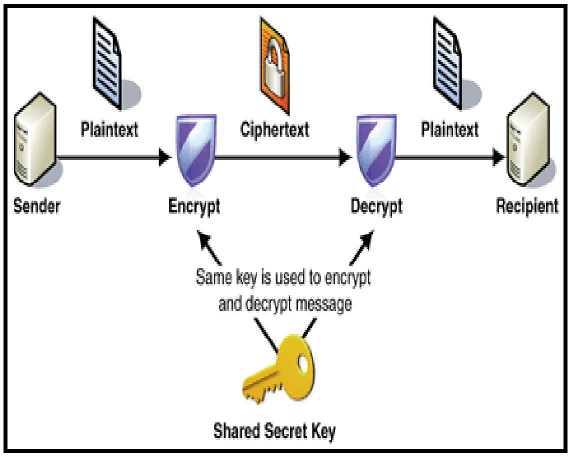


Figure 2: Symmetric Encryption and Decryption [90].

5.1.1 Advanced Encryption Standard (AES) / RIJNDAEL

At the end of the 1990s, the U.S. National Institute of Standards and Technology (NIST) had organized a competition, as it aimed to develop a substitution for the DES algorithm. The winner, named in 2001, is the Rijndael (pronounced "rhine-doll") algorithm, which had gradually manifested itself as the new Advanced Encryption Standard. Rijndael integrates the SPN model (Substitution-permutation network) by adopting the Galois field operations in each round. Slightly sharing a resemblance to RSA modulo arithmetic operations, the Galois field operations have been demonstrated as rather nonsensical, but they can be inverted in a mathematical manner. By nature, the AES' Security is not absolute, where it depicts a correlation between time and cost. Any questions raised about the encryption security should be along the lines of how long and how costly it will be for an attacker to discover a key.

Currently, it has been hypothesized that military intelligence services potentially have the technical and economic revenues to attack keys equivalent to about 90 bits, although any ordinary researcher has actually had any kind of exposure, of such a capability. The actual systems have demonstrated that today, within the limits of a commercial budget of about 1 million dollars, it can administer key lengths of approximately 70 bits. A rough estimate on the rate of technological advancement is expressed within the assumption that technologies will doubly increase the speed of computing devices annually at a static cost. If it is accurate, in theory, 128-bit keys would be in the range of a military budget in 30-40 years' time. To illustrate this, the current status for AES is shown here, where we presume an attacker is capable of building or purchasing a system that computes keys at one billion keys per second. At the very least, this is 1000 times faster than the fastest personal computer ever sold in 2004. Under this unfounded premise, the attacker will require about 10 000 000 000 000 000 000 000 years to try all potential keys for the version with prominent weakness, AES-128. Thus, the key length should be selected after making the decision about how long the security is required, and at what price it is to contain a secret key.

To date, there is no evidence that AES has any limitations in terms of launching any sort of attack other than performing rather thorough search, i.e. brute force, probable [13]. Even AES-128 has put forward large number of possible keys that are regarded sufficient, altogether implying the impracticality of such exhaustive search that takes up a very long time, provided no technological infiltration which leads to the computational power‘s availability and the theoretical study to increase drastically does not resort to any briefer procedure to skip having to perform the exhaustive search.

Relevant programmers need to be reminded of the variety of shortcomings, to steer clear from the time when the encryption comes into practice, and keys are produced. It is essential to make sure that every implementation is secure, but this is a tough call due to the fact that expertise would be needed to examine the implementation in detail and with great care. Any particular implementation should undergo an important aspect of assessment which is to make sure that such an examination has been conducted, or can be carried out [15] [16].

5.2 Asymmetric Cryptography

Asymmetric cryptography is a type of cryptography also known as public-key cryptography. Public-key cryptography is done by a pair of related keys. A message encrypted with a key can only be decrypted with the equivalent part of that key [17]. In public-key encryption, every participating party should have a pair of keys: a private one, which should be secured and known only by the holder, and a public one, which anyone can hold. If the encryption process is done with a party's public key, the decryption should be done with the counterpart private key. The inverse is also correct: if a message is encrypted with someone's private key, it should be decrypted with the user's public key [14].

In contrast to symmetric algorithms, asymmetric algorithms are new and recent. Most well-known one is RSA which was first invented in 1976 by Diffie and Hellman, and in 1977 Rivest, Shamir and Adleman introduced the RSA Cryptosystem, the first public-key system [17].

Public key Cryptography does not require a secure initial exchange of one or more secret keys to both sender and receiver. The asymmetric key algorithms are used to generate a mathematically linked key pair, a private key and a public key. The use of these keys allows the security of the authenticity of a message by producing a digital signature of a message using the private key, which can be verified using the public key. It also provides the protection of the confidentiality and reliability of a message. Public key cryptography is a crucial and widely-used technology around the world. It is an approach that has been employed by many cryptographic algorithms and cryptosystems. Some examples of well-known asymmetric algorithms include the RSA, ECC and NTRU [14] [13].

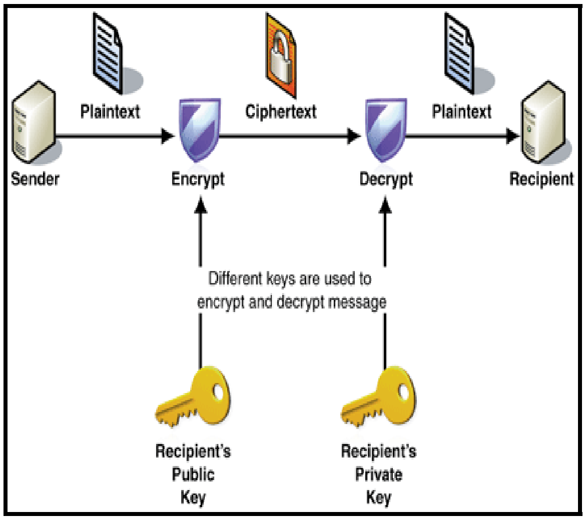


Figure 3: The process of asymmetric encryption [18]

5.2.1 NTRU Algorithm

Nth Degree Truncated Polynomial Ring Units (NTRU) was given the kick start in 1996 by three mathematicians: Jeffrey Hoffstein, Joseph H. Silverman, JillPipher. In 2009, NTRU Cryptosystem has received the endorsement to be systemized as standard by the Institute of Electrical and Electronics Engineers (IEEE) [19]. Being one of the widely known, robust cryptosystem algorithms, NTRU has been innovated to be presented as a novel cryptography generation that contributes to the enhanced performance of encryption and decryption processes that reflect a lot of cryptography-based problems. Despite the fact that the NTRU algorithm is still being progressed and that it necessitates further research to ensure perfection, it serves as a good alternative as a more solidified foundation for upcoming wireless communications because of several plus points: that the security is more assured, that it has great speed and the computational complications are reduced [12].

NTRU is a ring-based public key cryptosystem, because it leans on the dual ring operations of addition and multiplication. Bearing this in mind, it is noticeably dissimilar to the most widespread cryptosystems, which are group-based and where it uses only group operations to serve the parameters. The well-off arithmetical arrangement of the underlying ring is one advantage of the NTRU cryptosystem. Conversely, the ring structures in cryptography are not as thoroughly explored as the group theory, and security evidence is therefore is in for more convenient administration within groups [20].

By principal, the lattice-based systems and NTRU offer great speed, where they are also anticipated to endure successfully the advancement of fairly-sized quantum computers, because their root problems does not recognize any quantum algorithm, especially in general cases. It is also hard to suggest any secure instances, even if reference is made to a classical computing model. Moreover, it remains a fact that the complications that have surrounded the classical lattice reduction algorithm have not been grasped very well. [13] To date, there have been no established quantum algorithms that can unravel the lattice problems with more credible complexity than the classical algorithms. Therefore, the lattice-based schemes might show off their sense of survival in the quantum computation age [20].

5.3 Digital Envelope Method

Using secret-key cryptosystems, users must first agree on a session key, that is, a secret key to be used for the duration of one message or communication session. In completing this task there is a risk the key will be intercepted during transmission. This is part of the key management problem. Public-key cryptography offers an attractive solution to this problem within a framework called a digital envelope.

The digital envelope consists of a message encrypted using secret-key cryptography and an encrypted secret key. digital envelopes in this research uses public-key cryptography to encrypt the secret key in order to solve the problem of key exchange, In other words using asymmetric encryption to encrypt the keys shared and using symmetric encryption to encrypt the content of that transmission.

Suppose Alice wants to send a message to Bob using secret-key cryptography for message encryption and public-key cryptography to transfer the message encryption keys. Alice chooses a secret key and encrypts the message with it, then encrypts the secret key using Bob's public key. She sends Bob both the encrypted secret key and the encrypted message. When Bob wants to read the message he decrypts the secret key, using his private key, and then decrypts the message, using the secret key. In a multi-addressed communications environment such as e-mail, this can be extended directly and usefully. If Alice's message is intended for both Bob and Carol, the message encryption key can be represented concisely in encrypted forms for Bob and for Carol, along with a single copy of the message's content encrypted under that message encryption key [21] [22]. In the following table, the researcher lists a comparison of the proposed solution and similar implementation of NTRU and AES, in order to make the image closer to the reader for the advantages of both solutions. Noting that literature review is lack to such implementation except one performed on mobiles by [23].

Table 1: Comparison between this research implementation and similar implementation

Comparison Point	**Similar pervious implementation of NTRU+AES**	**Our proposed**
Architecture	P2P	Client-Server
Medium applied on	GSM	GSM or Internet connection
Confidentiality	Applicable	Applicable
Authentication	NA	Applicable
Suitability	Suitable for SMS	Applicable for email security (PGP or S/MIME), enhance the POP3, m-government e-commerce, SMS/MMS, instant messages such as Viber, WhatsApp.
Performance	Original AES is utilized.	Light weight AES that is suitable with mobile and other application is utilized.
Registration	Not required	Registration at trusted server is required.
Key generation	Done at the device (added more computing complexity).	Done at the server.
Security issues	Man-in-the-middle attack has opportunity to hack the system, this is due to the lack of authentication during the key exchange.	Each person authenticate using fully encrypted communication while the client authenticate himself after that using permanent key in addition to the public key encryption.

5.4 Hybrid Technique (The Proposed Model)

An extensive study is performed for an assessment of the hybrid technique in order to justify the selection of this technique over other individual security algorithms. Since NTRU is considered one of the most powerful security algorithms with no security breach so far, and due to its suitability to be used with embedded systems, the researcher has tested NTRU performance on mobile devices separated from AES [24]. That means NTRU does the encryption of EMR on the server side and decrypts the content of EMR on smartphone using its pair Asymmetric keys.

In order to validate the results obtained, the performance of NTRU should be compared later to the performance of NTRU + AES as hybrid cryptography to examine the applicability of this combination on smartphones. The result of running NTRU on smartphone has proved a drawback of NTRU built that mentioned in the reference of NTRU developers [22]. This drawback which is called message expansion is actually a feature of NTRU that interfere with any approach of big sized data (e.g. EMR). Message expansion implies that any block of data processed by NTRU will be expanded as output ciphertext based on the equation [22]:

Plain Text Block $\mathbf{N \log_2 \mathit{p}}$ **bits** **(1)**
Encrypted Text Block $\mathbf{N \log_2 \mathit{q}}$ **bits** **(2)**

Where q is always considerably larger than p, and one of the typical value set of (N, p, q) is (167, 3, 128. On the other hand NTRU has other drawback in which it affects the time consumed in encryption and decryption. NTRU processes the data by blocks of maximum 246 bytes (in the implementation used by our experiment), and large data files with even few kilobytes would need to be processed as multiple blocks and thus time will be consumed. Based on what stated, the NTRU is more suitable to digital envelope method by encrypting the keys shared in cryptography, rather than large message contents.

The proposed hybrid technique (based on the digital envelop concept introduced in section 5.3) makes use of both symmetric and asymmetric cryptography to come out with a better solution in processing data. Hence this technique is a suitable solution for securing EMR accessed or sent to users on smartphones, knowing that EMR sometimes shared with big size images or patient details.

Figure 4 below is a plain demonstration of digital envelope method that was adopted in this research in order to present the process of exchanging EMR between the server and the user, and to simulate the mechanism of encrypting and decrypting data during transmission.

The upper part of the figure represent the server side that does the encryption on EMR using the public key (asymmetric encryption) to encrypt the user's key, in same time using secret key (symmetric encryption) to encrypt the EMR, after that combined together with message digest to add more security on the content.

In the lower side of the diagram it demonstrates the decryption process which is performed on the user side (smartphone application).

In the Medical sector, doctors, patients and nurses need to have access to the medical records efficiently and in a secure manner. As information technology gets increasingly deployed in the medical sectors it becomes eminent that the medical records are stored electronically [30]. Secure Electronic Medical Records (SEMR), which aims at providing a set of services which will provide secure and efficient access of the EMRs to the patients, doctors, nurses and insurance agents. The set of services that are provided by SEMR include Authentication, Authorization and Secure communication. The researcher has chosen EMR as a case study for the reason of it relevancy and significance in recent security application. Furthermore it will provide convenient solution to smartphone user in order to access or explore their medical records anytime and anywhere.

Authentication involves identifying whether the user is a valid entity in the system. In order to access any resource in the system the user has to be authenticated first. One of the ways to implement the authentication service would be to ask for user identification in terms of username and password.

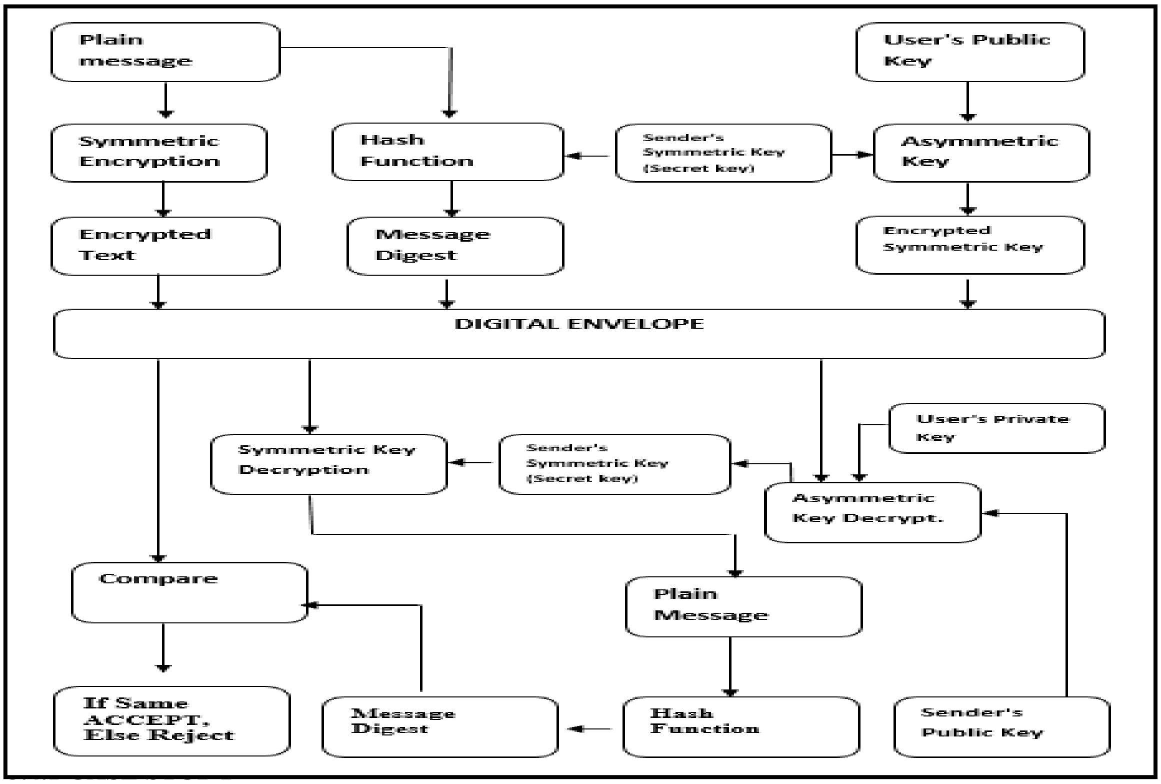


Figure 4: Digital Envelope Approach

Confidentiality involves identifying the privacy of the users and protection from illegal access. Authenticated users should access only those resources to which they have access to. Also, the type of access (read, write, execute) determine the capability of the user to access that resource. This functionality can be implemented by creating access control list for the resources. These ACLs identify the type of access the user has to the resource.

In a system like the medical sector which is a distributed in nature messages and data needs to be communicated amongst the different services. Thus it is important that the communication takes place securely. Such secure communications can be implemented by encrypting the data that is sent over the wire. In this section, the researcher has chosen EMR as a branch of m-Government categories to be applied as a case study, in order to have a simplified scenario on the process of hybrid technique performance. The mission in this case is to secure the transmission of EMR between hospital server and smartphone users; it also demonstrates secure access of EMR by users. Figure 5 shows demonstration graph of the processing flow.

5.4.1 Research Question

The following questions will give a better insight on the direction and the outcomes of the research. Based on the study, some important points have been presented in the earlier sections:

i. What issues tend to arise when m-Government system services are not transmitted in a secure manner?

ii. Is it applicable to use hybrid cryptography on mobile?

iii. Is it authentication and confidentiality verified by the proposed technique?

iv. Does the performance test support the security of the proposed system?

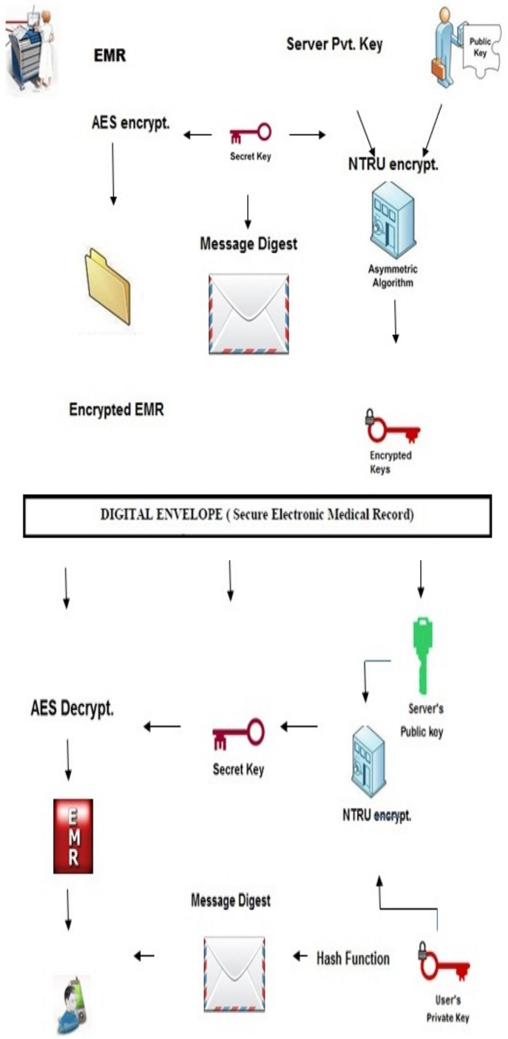


Figure 5: Illustration diagram of EMR flow

5.5 Data Collection Tools

In this section the researcher discusses the research tools used to obtain the results of the study. The system implementation runs using required tools that fulfill the necessary requirement to do implementation.

5.5.1 Hardware Tools

5.5.1.1 Desktop Computer

Dell Inspiron Desktop, with Intel® Core™ i3 Processor and a memory capacity of 4GB. This desktop features built-in wireless networking in order to provide connection to the Internet without wires. It is used as a main part in the implementation to simulate the process transmitting data between the server and the user, as the desktop represents the server side.

5.5.1.2 Smart Phone

Is a touchscreen Android smartphone designed, developed, and marketed by Samsung Electronics. It is used as a core part in the implementation in order to simulate the process of transmission between government and users. The process of implementation applied on three versions of the phone (i.e., S Plus, S II, and S III). The reason behind choosing this specific type of smartphone returns to its popularity in the android market as numerous users around the world purchasing it and easy to find samples to be applied in research experiment. Samsung Galaxy USB port is used to transfer the data between the phone and PC.

5.5.2 Software Tools

5.5.2.1 Eclipse

Eclipse is a multi-language integrated development environment written mostly in Java. The JDT project provides the tool plug-ins that implements a Java IDE which supports the development of any Java application, including Eclipse plug-ins. It adds a Java project nature and Java perspective to Eclipse. The researcher used the version of Juno Service, Release (2) to develop the code that used later as Android application.

5.5.2.2 Netbeans

An open source integrated development environment. NetBeans IDE supports development of all Java application types including JavaFX, Java ME, web, EJB and mobile applications. The version used is 7.1.2, operated on windows 7. The functions used are to measure encryption and decryption speed on desktop computer, and to connect the mobile with the PC throughout the server.

5.6 System Implementation

Developing this implementation carries the aim of proving the hybrid technique's applicability as an appropriate solution for protecting m-Government data transmissions based on the concept of digital envelop. For mobile implementation, we have used Android SDK, Java libraries and the Eclipse development environment to develop the system. For an improved understanding on the way the system works, the system is divided into three stages: stage one handles how the user can create a new account using the system. The second stage involves account activation as to provide a secret key for the user. The final stage includes the encryption, decryption and the manner in which the user can access or obtain the medical record.

The proposed system follows the client-server model, where the client is assumed to be a mobile device (Android smartphone, running an Android app), and the server is a plain Java application running on a traditional PC.

Stage 1: Creating Account

There are two types of users in the system: staff and subscriber. At one stage of the system, all users have to be registered with the System Registration Authority (SRA). The user has to provide an ID and mobile number as proof that he or she is given the right to access the system. The Administrator is responsible to examining the user identity and gathering all the necessary information about the user, and then making a new account for the user. Administrator will then provide the new user with an ID and a random impermanent password. The system will generate this random password with thirty-two alphanumeric characters. This password would be required to later activate the user account and create the user key.

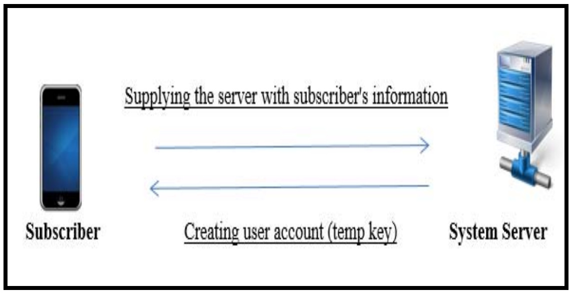
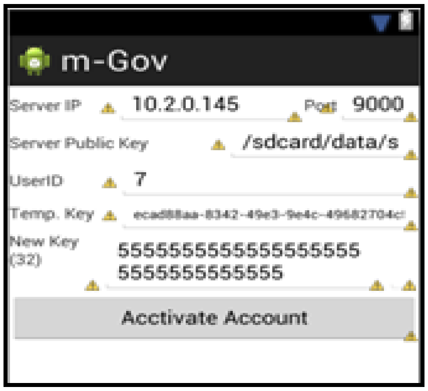

Supplying the server with subscriber's information
Subscriber
Creating user account (temp key)
System Server

Figure 6: Process of creating an account

Stage 2: User Account Activation

The subscriber will not have any access to the government's data before the activation of the account. Upon activation, the user sends to the server the temporary key, obtained during the account creation, along with his new password that is not known by anyone else. The temporary and new keys are encrypted using the server’s public key and NTRU algorithm. Finally, user adds his user ID with the activation request to the encrypted message and then he or she delivers it to the system server. The server will receive the user activation request. To decrypt the encrypted request, the server uses its private key with the NTRU algorithm. Then, following the user ID, server uses the temporary password to authenticate the user. If the decrypted temporary key matches the initial user key stored at the server, then the new password retrieved from the encrypted message is used for all later user authentication. At this time the server generates NTRU keys which are both public and private by nature. The server saves the public key with the user ID in its keys dictionary and encrypts the private key via AES Rijndael with the new password obtained from the encrypted message that the server has received from the user. At the server, the private key for the user is not saved. By contrast, on the user side, user will receive the encrypted message which will be decrypted using the AES Rijndael algorithm with his secret password. The figure below shows the user account activation graphical user interface with sample activation request.

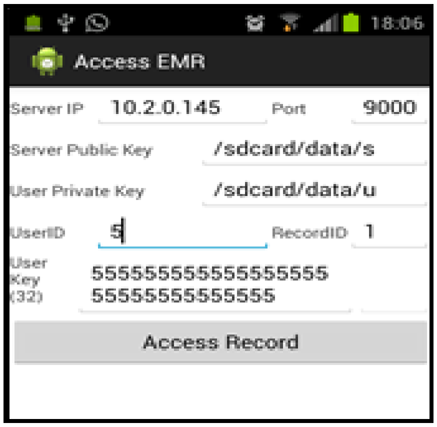


Server IP 10.2.0.145 Port 9000
Server Public Key /sdcard/data/s
UserID 7
Temp. Key ecad88aa-8342-49e3-9e4c-49682704c!
New Key (32) 5555555555555555555 5555555555555

Figure 7: User account activation GUI

Stage 3: Accessing Government Data

After account activation, the user is able to use his or her account to send a query to the system. For subscribers, a user can send a query in regard of his own data only. Nevertheless, the server still has to check the users’ authority should they need to access certain records. To send a query to the system server for a data message, the user sends a request in an encrypted message. The server then checks the user’s authority in reference to the user’s details, for example user's mobile number. If the user is confirmed authorized, the server encrypts the government data by using AES Rijndael with the user’s secret key and it will encrypt the keys using NTRU. Then the server will send it to the user. For the user, when the encrypted message is received, user uses the NTRU with his private key for the key decryption. The decryption is done using AES Rijndael with the secret key to obtain the m-Government data. The figure below shows the Accessing m-Government Graphical user Interface.

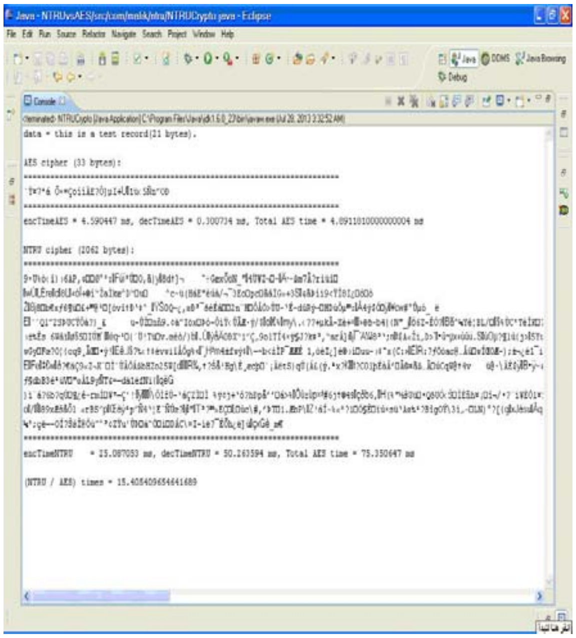


Server IP 10.2.0.145 Port 9000
Server Public Key /sdcard/data/s
User Private Key /sdcard/data/u
UserID 5 RecordID 1
User Key (32) 555555555555555555 55555555555555

Figure 8: Accessing Government Data GUI

5.6.1 Results Interface

The screenshot in figure 9 shows the result of encrypting different size of data using NTRU and AES separately.


Java - NTRUvsAES/src/com/malik/ntru/NTRUCrypto.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help
Java DDMS Java Browsing
Debug
Console

```
<terminated> NTRUCrypto [Java Application] C:\Program Files\Java\jdk1.6.0_27\bin\javaw.exe (Jul 28, 2013 3:32:52 AM)
data = this is a test record(21 bytes).

AES cipher (33 bytes):
*****************************************************************************
[illegible]
*****************************************************************************
encTimeAES = 4.590447 ms, decTimeAES = 0.300734 ms, Total AES time = 4.891181000000004 ms

NTRU cipher (2062 bytes):
*****************************************************************************
[illegible]
*****************************************************************************
encTimeNTRU    = 25.087053 ms, decTimeNTRU = 50.263594 ms, Total AES time = 75.350647 ms

(NTRU / AES) times = 15.405409654641689
```

Figure 9: Result of encrypting different size of data using NTRU and AES separately

The experiment then is performed on larger size of data. Table 2 presented the outcome of different file sizes of EMR after the encryption is performed. As noticed in the table, first file used bytes unite because it is small file compared to the other two files followed in the table. We used different sizes of data in order to achieve consistency, as EMR transmitted to the user with different sizes and AES carries the duty to encrypt and decrypt EMR based on digital envelope method.

Table 2: Different file sizes of AES encryption

Data	algorithm	Output	Time m.s.
File 1 (21 B)	AES	33 Byte	4.891
File 2 (1.76 KB)	AES	1.88 KB	8.843
File 3 (38.08 KB)	AES	38.08 KB	24.921
File 4 (1.20 MB)	AES	1.20 MB	365.734
File 5 (2.03MB)	AES	2.03 MB	581.309

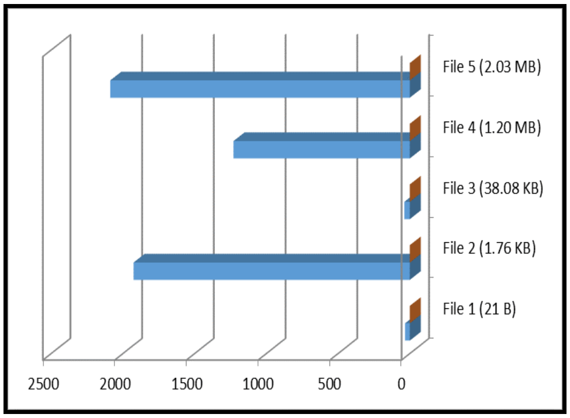
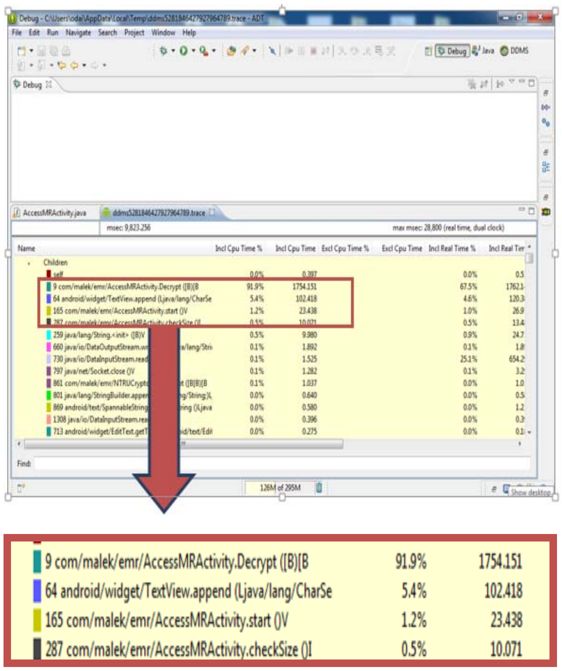


Figure 10: Different amounts of encrypted data produce different amounts of time

Figure 10 demonstrates how different data sizes of the EMR are carried by different speeds during encryption. The blue bar which represents the data output and then the red bar which implies the time consumed in milliseconds.

5.6.2 Hybrid Technique Using NTRU+AES

In this section, we present the results obtained as the system is operated. The output of results is obtained through Eclipse environment. The screenshot below in figure 11 shows the result of NTRU+AES performance on the smart phone while a record on the server is being accessed.


Debug - C:\Users\odai\AppData\Local\Temp\ddms5281846427927964789.trace - ADT
File Edit Run Navigate Search Project Window Help
Debug
Java
DDMS
AccessMRActivity.java
ddms5281846427927964789.trace
Find:
126M of 295M
Show desktop

msec: 9,823.256 max msec: 28,800 (real time, dual clock)

Name	Incl Cpu Time %	Incl Cpu Time	Excl Cpu Time %	Excl Cpu Time	Incl Real Time %	Incl Real Tir
Children						
self	0.0%	0.397			0.0%	0.5
9 com/malek/emr/AccessMRActivity.Decrypt ([B)[B	91.9%	1754.151			67.5%	1762.1
64 android/widget/TextView.append (Ljava/lang/CharSe	5.4%	102.418			4.6%	120.3
165 com/malek/emr/AccessMRActivity.start ()V	1.2%	23.438			1.0%	26.9
287 com/malek/emr/AccessMRActivity.checkSize ()I	0.5%	10.071			0.5%	13.4
259 java/lang/String.<init> ([B)V	0.5%	9.980			0.9%	24.7
660 java/io/DataOutputStream.w a/lang/Stri	0.1%	1.892			0.1%	1.8
730 java/io/DataInputStream.read	0.1%	1.525			25.1%	654.2
797 java/net/Socket.close ()V	0.1%	1.282			0.1%	3.2
861 com/malek/emr/NTRUCrypt t ([B[B)[B	0.1%	1.037			0.0%	1.0
801 java/lang/StringBuilder.apper ng/String;)L	0.0%	0.640			0.0%	0.5
869 android/text/SpannableString ring (Ljava	0.0%	0.580			0.0%	1.2
1308 java/io/DataInputStream.rea	0.0%	0.396			0.0%	0.3
713 android/widget/EditText.getT id/text/Edi	0.0%	0.275			0.0%	0.1

9 com/malek/emr/AccessMRActivity.Decrypt ([B)[B	91.9%	1754.151
64 android/widget/TextView.append (Ljava/lang/CharSe	5.4%	102.418
165 com/malek/emr/AccessMRActivity.start ()V	1.2%	23.438
287 com/malek/emr/AccessMRActivity.checkSize ()I	0.5%	10.071

Figure 11 : Eclipse DDMS functions records

As shown in figure 11 above the first column is the inclusive time spent in the method when the function is requested. The second column is the total time from the start to end of the process; it includes the waiting time for the process to be executed. This indicates that whenever the user presses the access button, the profiling method will start to calculate the decryption time required to decrypt the cipher text, or encrypt the file sent to the user by the server.

5.6.2.1 Results of NTRU and AES performance on Smartphones

First experiment conducted has given us the results listed in the table below:

Table 3: NTRU and AES first running

Mobile	NTRU performance		AES performance		AES \NTRU
	% of Incl time	Execution time in ms	% of Incl time	Execution time in ms	
Samsung G S 1	93.6 %	1226 4.374	0.04 %	5.646	0.04%
Samsung G S II	90.5 %	1482. 910	0.05 %	0.855	0.05%
Samsung G S III	91.0 %	1779. 724	0.14 %	2.777	0.15%

As seen in table 3, Incl (inclusive) CPU time is the time for which the process uses the CPU, where Inclusive time is the time spent in the method plus the time spent in any called-upon functions. The CPU percentage rates indicate that the NTRU dominates the processing during the decryption. However, the lower part is occupied by AES decryption which is below 0.2% and in very small amount of milliseconds. Experiment is repeated on the same data to ensure that the performance of hybrid encryption can be examined consistently.

The second experiments have revealed the results listed in table 4, showing only slight difference to the results in the experiment done earlier.

Table 4: NTRU and AES second running

Mobile	NTRU performance		AES performance		AES \NTRU
	% of Incl time	Execution time in ms	% of Incl time	Execution time in ms	
Samsung G S 1	96.8 %	12230.866	0.04%	5.615	0.04 %
Samsung G S II	91.4 %	1486.572	0.05%	0.915	0.06 %
Samsung G S III	90.7 %	1796.387	0.12%	2.441	0.13 %

As we notice in the table above, the rate of Incl CPU time of NTRU had increased by 3% in the first version compared to the first experiment, the second version saw another increase by 1% only compared to the first experiment, and in the last version it decreased with 0.3% only. In the meantime, the AES rates shows slight difference in milliseconds compared to the previous running. In the last experiment the researcher has obtained these results, as shown in table 5.

Table 5: NTRU and AES third running

Mobile	NTRU performance		AES performance		AES \NTRU
	% of Incl time	Execution time in ms	% of Incl time	Execution time in ms	
Samsung G S 1	96.1 %	12235.443	0.03 %	4.364	0.03%
Samsung G S II	90.3 %	1487.732	0.15 %	2.411	0.16%
Samsung G S III	92.3 %	1787.628	0.1%	2.808	0.15%

Again in experiment number 3, we notice a small variance in version one (Samsung S 1) and version 3 (Samsung S III), however there is higher score of AES time in version 2 (Samsung S II) which leads to the decreased ratio between NTRU time and AES time from 0.06% in the previous running to 0.16% in the last running. This change would occasionally take place due to varying CPU load incurred by other active apps. The ratios shown in the tables above validate the fact that AES is much faster than NTRU.

5.6.3 Relative Performance of Algorithms on Mobile Devices Compared to PC

5.6.3.1 NTRU Activation process (Encryption) on PC

As the activation process has to be performed on the server side, the researcher has examined the performance of the NTRU decryption on PC using NetBeans environment. The result of the test is included in the screen shot below:

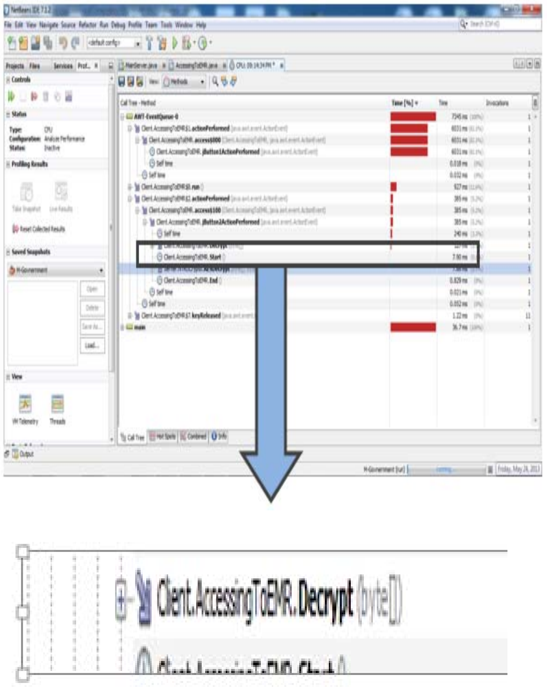
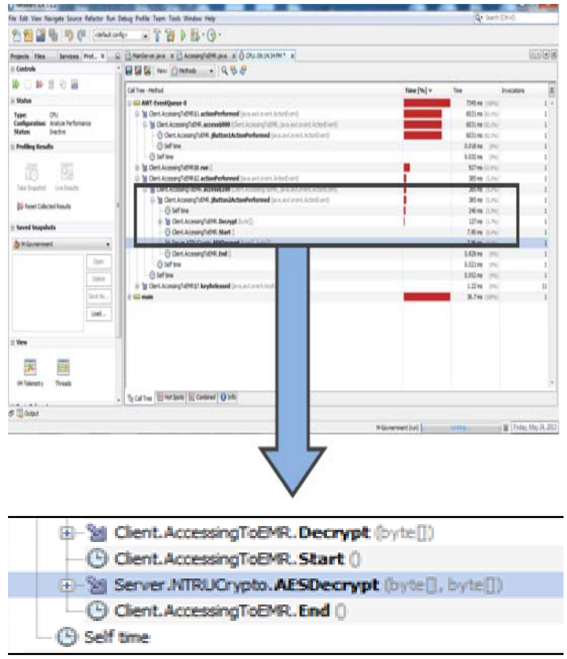


Figure 12: NTRU decryption running on NetBeans

5.6.3.2 AES Access Activity on PC

We note that there is an advantage on the performance of AES decryption on PC as compared to that on mobile. The researcher in this section includes the result of accessing government data throughout the PC, in order to establish the speed difference between AES on PC and AES on mobile. The figure below illustrates the AES Decrypt performance on PC using NetBeans.



```
⊞ Client.AccessingToEMR.Decrypt (byte[])
  Client.AccessingToEMR.Start ()
⊞ Server.NTRUCrypto.AESDecrypt (byte[], byte[])
  Client.AccessingToEMR.End ()
Self time
```

Figure 13: AES decryption running on NetBeans

The table 6 below lists the values obtained for both NTRU and AES on both machines- the Mobile and PC. The first row which presents PC results counts the ratio of NTRU to AES which is 16.2 times and this supports the fact cited on the speed of the AES compared to the NTRU as mentioned in some papers [24] [25], that AES is about 18-20 times faster than the NTRU. The Mobile or smart phone-derived results in the second row demonstrate a more lagged performance, in view of the fact that smart phones compared to PCs are smaller, and have slower processing time. The ratio of NTRU to AES in mobile is 2166 in this experiment, which is a new value to be scored for mobile cryptography.

Table 6: NTRU and AES comparison between Mobile and PC

Platform	NTRU time (ms)	AES time (ms)	NTRU / AES
PC	127	7.86	16.2
Mobile	12230.866	5.646	2166

5.7 Research Discussion And Contribution

The previous sections have attempted to come up with a form of enhancement on the m-Government security practices. It introduces a new solution for protecting the m-Government transmission and accounts for EMR as a case study as it checks on the proposed solution's performance. The research contributes in leveraging the potential use of smart phone devices in administering sensitive data communication. In this section, the researcher links the work that has been done with the objectives of this research, and explains how the objectives have been carried out.

The first objective is to examine a number of security algorithms that have been adopted in m-Government. Security algorithms tend to play a pivotal role in any security project, and an excellent tool which aids people in protecting their personal information is the use of cryptography. Leaning on this concept, it is crucial to consult some well-known algorithms used in the field of security and demonstrate their propensity. Some of these algorithms include the AES, RSA, ECC, and NTRU. A very careful study on these algorithms has narrowed the number of algorithms that will form the hybrid technique, and they are the AES-Rijndael and NTRU.

The second objective is to create and enforce a hybrid technique of AES and NTRU algorithms on mobile devices that falls within the m-government framework. This objective intends to analyze and assess multiple m-Government implementations and platforms from the security scope, in order to define what constitutes the best practice for m-Government's security and reliability, also to try to meet the aim of securing m-Government transmission through varying cryptography tools. In addressing this, the researcher has conducted several surveys where the contributions made in the m-Government area are listed, and to stand on, and be appreciative of, the advantages and disadvantages brought along by every implementation. An analysis study on cryptography algorithms has also been carried out, where all the strengths and vulnerabilities of each algorithm are highlighted and thus bringing us to the justification as to why the hybrid technique is proposed. In this objective the implementation of the hybrid technique is elaborated, with close reference to the practical analysis of AES and NTRU. It tells us how the hybrid technique comes about, using the digital envelope technique to benefit from both symmetric and asymmetric encryptions, and yet with minimum loss.

The last objective is to form a sort of appraisal on the hybrid technique proposed and to shed light on the proponents' view of the applicability of the implementation. A lot of efforts have been exerted on justifying every step of the implementation for ease of assessment in the future. The implementation quality relies on a progress-monitoring approach. A number of runs had been done using different versions of smartphone in order to obtain results that are closer to accuracy. The assessment of this technique has been proven in NTRU + AES performance, compared to the performance of NTRU alone where its issue concerning message expansion has yet to be dealt with. Another form of feasible evaluation is to develop a smart phone application which enables this technique to be applied on the actual environment of m-Government- this may be seen as a more valid way in which the performance can be examined.

5.8 Research Limitations

The limitation of the current research emerges when it comes to the protocol application; the new m-Government application was used in a simulated environment although the entire tests have been run on real mobiles. As we respect the privacy of the m-government data and acknowledge the fact that any disclosure might hamper this privacy, the system had to be executed with simulated data.

Another limitation lies in the fact that the new solution is developed for Android OS, while the other OSS like iOS do not support the new software. Theoretically, our solution is applicable for iOS and other OSS, however, the phenomena of the research has channeled the solution into only a particular environment.

6. CONCLUSION

This work have attempted to come up with a form of enhancement on the m-Government security practices. It introduces a new solution for protecting the m-Government transmission and accounts for EMR as a case study as it checks on the proposed solution's performance. The research contributes in leveraging the potential use of smart phone devices in administering sensitive data communication. The first objective was to examine a number of security algorithms that have been adopted in m-Government. Whereas the second objective was to create and enforce a hybrid technique of AES and NTRU algorithms on mobile devices that falls within the m-government framework. This objective intends to analyze and assess multiple m-Government implementations and platforms from the security scope, in order to define what constitutes the best practice for m-Government's security and reliability, also to try to meet the aim of securing m-Government transmission through varying cryptography tools. The assessment of this technique has been proven in NTRU + AES performance, compared to the performance of NTRU alone where its issue concerning message expansion has yet to be dealt with. The hybrid technique of AES and NTRU cryptography algorithms is found to be able to withstand malicious attacks if transmission takes place on unprotected channels. The results are obtained through a real time experiment run in just few times, and in order to look for the explanation for this phenomena, it is perhaps best if we could make an experiment on the hardware structure or software programs that have an effect on the encryption process, where this is for future research to work on.

REFERENCES:

[1] K. Karan and M. C. H. Khoo, "Mobile Diffusion and Development: Issues and Challenges of M-Government with India in Perspective," Proceedings of M4D, 2008.

[2] O. Ostberg (2012. September 10) The Potsdam e-Government Competence Center: Mobile Government Swedish Style [Online]. Available: http://www.ifg.cc/index.php?option=com_content&task=view&id=6184&Itemid=92

[3] T. Zefferer, "Mobile Government," Austria Secure Information Technology Center, 2011.

[4] OECD-ITU "M-Government: mobile technologies for responsive governments and connected societies" Organisation for Economic Co-Operation and Development, 2010.

[5] W. Abramowicz, L. Karsenty, P. M. Olmstead, G. Peinel, D. Tilsner, and M. Wisniewski,"USE-ME. GOV (Usability-driven open platform for Mobile government)," 2005.

[6] O. f. E. Co-operation and Development, "M-government: Mobile Technologies for Responsive Governments and Connected Societies, " OECD, 2011.

[7] Mani, Subramani, et al. "Differential diagnosis of dementia: A knowledge discovery and data mining (KDD) approach." Proceedings of the

AMIA Annual Fall Symposium. American Medical Informatics Association, 1997.

[8] Raghupathi, Wullianallur, and Someswar Kesh. "Interoperable electronic health records design: towards a service-oriented architecture." E-service Journal 5.3, 39-57, 2007.

[9] Miller, Robert H., and Ida Sim. "Physicians' use of electronic medical records: barriers and solutions." Health affairs 23.2, 116-126, 2004.

[10] Journal of medical science (August 2013) [Online]. Available: http://www.jmir.org/index

[11] A. S. K. Pathan, H. W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," 2006, pp. 6 pp.-1048.

[12] B. Schneier and P. Sutherland, Applied cryptography: protocols, algorithms, and source code in C: John Wiley & Sons, Inc., 1995.

[13] S. K. YADAV, "Some Problems in Symmetric and Asymmetric Cryptography," Ph.D. Dissertation, Dept. of mathematics, Agra Univ., 2010.

[14] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography: CRC, 1996.

[15] A. Naji, S. A. Hameed, B. Zaidan, W. F. Al-Khateeb, O. O. Khalifa, A. Zaidan, and T. S. Gunawan, "Novel Framework for Hidden Data in the Image Page within Executable File Using Computation between Advanced Encryption Standard and Distortion Techniques," arXiv preprint arXiv, 0908.0216, 2009.

[16] H. O. Alanazi, A. Jalab, A. Zaidan, and B. Zaidan, "New frame work of hidden data with in non-multimedia file," Int. J. Comput. Network Security, vol. 2, pp. 46-54, 2010.

[17] H. Alanazi, M. L. M. Kiah, A. Zaidan, B. Zaidan, and G. M. Alam, "Secure topology for electronic medical record transmissions," Int. J. Pharmacol, vol. 6, pp. 954-958, 2010.

[18] J. Hogg, Web service security: Scenarios, patterns, and implementation guidance for Web Services Enhancements (WSE) 3.0: Microsoft Press, 2005.

[19] J. Hoffstein, J. Pipher, and J. Silverman, "NTRU: A ring-based public key cryptosystem," Algorithmic number theory, pp. 267-288, 1998.

[20] D. AZTEC, "IST-2002-507932 ECRYPT," Information Society Technologies, 2002.

[21] EMC Corporation (June 2013) RSA laboratories [Online]. Available: http://www.rsa.com/rsalabs/node.asp?id=2184

[22] Hoffstein, Jeffrey, Jill Pipher, and Joseph H. Silverman. "NTRU: A ring-based public key cryptosystem." Algorithmic number theory. Springer Berlin Heidelberg, 267-288, 1998.

[23] Kiah, Mat, and Miss Laiha. "A novel peer-to-peer SMS security solution using a hybrid technique of NTRU and AES-Rijndael." Scientific Research and Essays2.22, 3455-3466, 2010.

[24] Hermans, Jens, Frederik Vercauteren, and Bart Preneel. "Speed records for NTRU." Topics in Cryptology-CT-RSA 2010. Springer Berlin Heidelberg, 73-88, 2010.

[25] Al-Bakri, Sameer Hasan, et al. "Securing peer-to-peer mobile communications using public key cryptography: New security strategy." International Journal of the Physical Sciences 6.4, 930-938, 2011.

[26] Kim, Yoojung, et al. "Architecture for implementing the mobile government services in Korea." Lecture notes in computer science, 601-614, 2004.

[27] Griffin, David, Philippa Trevorrow, and Edward Halpin. "Using SMS texting to encourage democratic participation by youth citizens: A case study of a project in an English local authority." Asymptotic and Computational Methods in Spatial Statistics, 102, 2009.

[28] Coordinating, N. E. C. "M-Government: Theconvergence of wireless technologies and e-Government". Retrieved May 21, 2013, from www.ec3.org/Downloads/2001/m_Government_ED.pdf

[29] Usman, M., M. A. Jan, and X. S. He. "Cryptography-based Secure Data Storage and Sharing Using HEVC and Public Clouds." Information Sciences, 2017.

[30] Wolff, Jennifer L., et al. "Inviting patients and care partners to read doctors' notes: OpenNotes and shared access to electronic medical records." Journal of the American Medical Informatics Association 24.e1, e166-e172, 2017.

[31] Singh, Mr S. Pratap, and M. Ekambaram Naidu. "A Review: Performance analysis of various Cryptographic Symmetric Algorithms." 2017.