Apr - Jun © 2016 Transactions; 3(10): 427-433

International Journal of Research and Applications

ISSN (online): 2349-0020 ISSN (print): 2394-4544 http://www.ijraonline.com/

Research Article

A Security Protocol for mobile-banking and payment using SMS and USSD in Ethiopia

[1] Ramesh Gadde, [2] Kifle Berhane. N and [3] Fthi Arefayne Abadi

Corresponding Author:
gadde.padma22@gmail.com

DOI:
http://dx.doi.org/10.17812/IJRA.3.10(72)2016

Manuscript:
Received: 9th Apr, 2016
Accepted: 15th May, 2016
Published: 28th June, 2016

Publisher:
Global Science Publishing Group, USA
http://www.globalsciencepg.org/

ABSTRACT

Short message service (SMS) and Unstructured Supplementary Services Data (USSD) are a very popular and easy to use communications technology for mobile phone devices. Originally, these services were not designed to transmit secured data, so the security was not an important issue during its design. Yet today, it is widely used to exchange sensitive information between communicating parties i.e. HelloCash, Ethio Gebeta, Lehulu, CBE M-banking, 8100, 8400 and so much more. Due to the vulnerable nature of SMS and USSD this paper proposes an alternative solution that provides a client-server SMS and USSD security protocol that guarantees provision of confidentiality, authentication, integrity, non-repudiation, and file compression security services. A hybrid cryptographic scheme is used which combines the Identity Based Encryption (IBE) and AES-Rijndael algorithms without key distribution servers and certificate authorities to achieve more robust functionality. HMAC-SHA256 hashing algorithm will be used to generate a message digest. IBE will be used to digitally sign the message and to encrypt the encryption key used on AES. LZW compression will be used to compress the SMS. Unlike any previous works that involve certificate authority and key management, this protocol is proposed to be used in mobile banking and payment once a user successfully subscribes to the service.

Keywords: USSD, HelloCash, IBE, HMAC-SHA256.

[1 2 3] Department of CSE, Mekelle Institute of Technology, Mekelle University, Tigray, Ethiopia.

IJRA - Year of 2016 Transactions:
Month: April - June
Volume – 3, Issue – 10, Page No's: 427-433
Subject Stream: Computers

Paper Communication: Author Direct

Paper Reference Id: IJRA-2016: 3(10)427-433

DOI: 10.17812/IJRA.3.10(72)2016

International Journal of Research and Applications
April-June © 2016 Transactions 3(10): 427-433

eISSN : 2349 – 0020
pISSN : 2394 – 4544
www.ijraonline.com

COMPUTERS RESEARCH ARTICLE

A Security Protocol for mobile-banking and payment using SMS and USSD in Ethiopia

[1] Ramesh Gadde, [2] Kifle Berhane. N and [3] Fthi Arefayne Abadi

[1 2 3] Department of CSE, Mekelle Institute of Technology, Mekelle University, Tigray, Ethiopia.
gadde.padma22@gmail.com, abemuludani@gmail.com and fthiabadi@gmail.com

ABSTRACT

Short message service (SMS) and Unstructured Supplementary Services Data (USSD) are a very popular and easy to use communications technology for mobile phone devices. Originally, these services were not designed to transmit secured data, so the security was not an important issue during its design. Yet today, it is widely used to exchange sensitive information between communicating parties i.e. HelloCash, Ethio Gebeta, Lehulu, CBE M-banking, 8100, 8400 and so much more. Due to the vulnerable nature of SMS and USSD this paper proposes an alternative solution that provides a client-server SMS and USSD security protocol that guarantees provision of confidentiality, authentication, integrity, non-repudiation, and file compression security services. A hybrid cryptographic scheme is used which combines the Identity Based Encryption (IBE) and AES-Rijndael algorithms without key distribution servers and certificate authorities to achieve more robust functionality. HMAC-SHA256 hashing algorithm will be used to generate a message digest. IBE will be used to digitally sign the message and to encrypt the encryption key used on AES. LZW compression will be used to compress the SMS. Unlike any previous works that involve certificate authority and key management, this protocol is proposed to be used in mobile banking and payment once a user successfully subscribes to the service.

Keywords: USSD, HelloCash, IBE, HMAC-SHA256.

1. INTRODUCTION

Even though Ethio-Telecom, the only internet service provider (ISP) in Ethiopia, provides very low quality services of internet it has offered SMS and USSD as an alternative tools for mobile banking. For instance, USSD services are used in HelloCash system to transfer money, pay in supermarkets and hotels [1], transferring mobile money, recharging your phone money and more and SMS services are used in Answer and Question (A&Q), fund rising, Lottery services, 8400, and 8100 which was used to collect around 33 million ETB from people across the country for the Construction of the Grand Ethiopian Renaissance Dam (GERD) [2]. Recently, an application called lehulu, powered by Kifiya Corporation, has begun to be used for paying electrical, water and telecommunication bills at one place and it has planned to start online payments[3].

Mobile banking system is one which provides all daily banking operations to customer with one click of his mobile handset with supported application. M-banking system has a potential to provide access or delivery of very specific and highly necessary information to customer. Mobile banking is a recently new research area. At present, many banks are promoting their mobile banking services heavily. As mobile banking becomes popular, the concern for security of mobile banking is raised.

There is perhaps no software engineering topic of more timely importance than application security. Attacks are costly, whether the attack comes from inside or out, attacks can expose any company to liability for damages. As computer (and especially Internet) technologies evolve, security attacks are becoming more sophisticated and frequent. Staying on top of the most up-to-date techniques and tools is one key to application security; the other is a solid foundation in proven technologies such as data encryption, authentication, and authorization. The growing number of programmers and hackers has

led to the raise of the following serious problems. First, several mobile applications are distributed to steal mobile money without the consent of the users. Second, lack of confidence on the transactions as a result of the insecure mobile banking services in Ethiopia.

Therefore, this paper primarily aims to study the current SMS banking encryption techniques and devise strong security protocol for secure transaction in mobile banking in Ethio-Telecom and other private and governmental companies. Additionally, by blocking attackers everywhere from Stealing user's mobile money, modifying packet on transmission this project plans to increase confidence of users and entrepreneurs on mobile banking.

2. BACKGROUND AND RELATED WORKS

The contents of SMS and USSD are visible to and monitored by anyone who tapped to the packets as they are transmitted as a plain text. The network provider itself i.e. Ethio Telecom which is generally regarded as insecure stores messages temporarily in servers until delivery making the contents and addresses vulnerable to ISP attacks. A hacker can easily hack the SMS center, base stations, and GSM servers and read what the SMS contains and what the USSD code is. We will now discuss the security attacks, available security mechanisms and security constraints. Most of the security attacks reside to the following four type of threats

a. Security threats

1. Man-in-middle Attack: the attacker can use a false BTS with the same mobile network code as the subscriber's legitimate network to impersonate himself and perform a man-in-the-middle attack. This also include masqueraders.
2. Message Disclosure: SMS and USSD are sent as plain text which allows full disclosure of the contents to outsider.
3. Denial of Service (DOS) Attacks: DOS attacks are made possible by sending repeated messages to a target mobile phone, making the victim's mobile phone inaccessible.
4. SMS Tapping: The attacker can tap an SMS in different places including from radio broadcast or base transceiver station (BTS). If the attacker has an access to the BTS or other parts of the GSM network, then the tapping is easy. The security services that can be used to counteract the security attacks are discussed below.

b. Cryptographic Security Mechanisms

Cryptography can be defined as the conversion of data into a scrambled code and then sending it to the recipient; the scrambled code can be decrypted to retrieve the original data once it is received. It has two main forms for encrypting data; symmetric and asymmetric encryption. Beside these two cryptography techniques there are several security technologies and mechanisms discussed below

1. **Symmetric key cryptography**: which is also called secret key cryptography. It is a type of cryptography where the same key is used to encrypt and decrypt the message.

2. ID-**based encryption**, or **identity-based encryption** (IBE): is an important primitive of ID-based cryptography. As such it is a type of public-key encryption in which the public key of a user is some unique information about the identity of the user (e.g. a user's email address).

3. **Message authentication**: is concerned with: protecting the integrity of a message, validating identity of originator and non-repudiation of origin (dispute resolution). An authenticator, signature, or message authentication code (MAC) is sent along with the message. Private Key ciphers or hash function can be used to generate an authenticator.

4. **Hashing functions**: are used to condense an arbitrary length message to a fixed size, usually for subsequent signature by a digital signature algorithm. They are one-way functions so that messages are not disclosed by their signatures.

5. **Message digests**. Coupled with message authentication codes, a technology that ensures the integrity of your message.

6. **Digital signatures**: Only the owner of the private-key can create the digital signature, hence it can be used to verify who created a message anyone knowing the public key can verify the signature (provided they are confident of the identity of the owner of the public key - the key distribution problem). Usually a hash of the message is signed instead of the whole message, because signing the whole message would mean doubling the size of information exchanged. Let's proceed to what types of security constraints should be achieved that will help our goals. All the security services mentioned above are used independently and with one another to satisfy the following vital security requirements.

c. Security constraints

In order to make SMS and USSD a secure medium of mobile banking we need to make sure following security constraints are met with minimum cost of overhead.

Confidentiality it prevents the unauthorized user to assess the private information. It can be met using the private key cryptography.

Integrity it is preventing anybody other than authorized parties from altering the SMS and USSD. It can be achieved by the use of digital signatures and hashing functions.

Non-repudiation it provides security service that prevent participant from denial of message transmission service. The receiver must prove the message is coming from authorized sender and the sender needs to be sure it is sending to authorized receiver. It is met only by using digital signatures with the help of hashing functions.

Authentication it gives assurance to the communication party that it claims to be. Message authentication functions will be used to authenticate a user. Enough is said about the technologies and services around security along with the threats and protections. We are left with the current security level in M-banking services in Ethiopia, security analysis of GSM network and what researchers has done so far. The following part discusses how the current M-banking in Ethiopia works along with its security shortfalls.

HelloCash: A person can register in the nearest bank branch or agent providing HelloCash Services and an Agent/bank teller initiates your HelloCash registration using their mobile phone. The helloCash system will call your mobile phone and ask you to Carefully select your 4-digits secret PIN for your helloCash account then you will receive an SMS to confirm your HelloCash account creation and you are good to transfer (*[number]#), withdraw (Call Short USSD No.), pay for services (Call USSD), and check balance (USSD call)[1]. Now, look how inconvenient it is to circulate value for hundreds of millions of households (money) that rely only on a 4-digits PIN that can easily be broken by a brute force attack. The same goes for the commercial bank of Ethiopia's (CBE's) mobile banking and for Kifiya's Lehulu water, electricity and telecommunications bill mobile payment [3].

Voting Singers (8400): In 2008 E.C (2016 September) there was a national Balageru idol singer's final competition and citizen's vote was given 60 percent value to decide a winner. It has been seen a 52 percent vote gap between the winner (first rank) and the second winner, when there was almost no/comparable difference among the others [15]. How confident was the voting process? Can there be a hacker involved? Well our guess is as good as yours Ethio-Telecom's services like this and others like, 8100 has no security measures at all. As stated above you have not been attacked does not imply a forest is a safe place to live.

Ethio Gebeta: This Ethio-telecom service is one of the latest package (phone air time or money) transactions being used in Ethiopia. It makes use of USSD and SMS to buy a package for week, a month or a day or to send packages of money for a friend or a family. Like any other Ethio-Telecom services, it has not made security an issue.

GSM security shortfalls

GSM network uses its own encryption algorithm when data is sent among several base transceiver stations (BTS) over the radio waves. It is transmitted in encrypted format by using A5 algorithm in which the attack to A5 algorithm is already known [4] and Ethio-telecom disables the encryption anyway to speedup communication by reducing security overhead. BTS also use A3/A8 authentication algorithm that is considered as the weakest authentication protocol [4]. Overall, a message is transmitted in unencrypted format from the mobile operator's network to the message center and then stored as plain text available for anyone who got access (including hackers) to the servers until it is delivered to its destination. There is no doubt of the need for application level security instead of trusting the GSM security protocols.

Previous researches

Many researchers have proposed solutions to secure the mobile phone SMS communication by using public key cryptography [5, 6, 7, and 11]. One of the main reasons for not implementing the standard public key cryptography in the current telecom architecture is the restricted resources (that is, user will be charged twice for key exchange session and the SMS alone) in the mobile phone devices. The second important reason is the user's authentication scheme and knowledge difference among users. Many researchers have also suggested the use of private key cryptography [12, 13 and14] which is not enough to meet the security constraints mentioned above.

Why not PGP (pretty good service)

PGP combines the advantages of both asymmetric and symmetric encryption, while also downplaying the disadvantages of both. PGP parties have each 2 keys one public and the other session or private. The session key is used to encrypt the message while the public key is used to encrypt the session key [8]. Even PGP will not guarantee non-repudiation or message integrity. Despite the failure to meet the criteria's set, PGP has problems with administering conflicting versions and compatibility Issues, complexity of use (requires training), no recovery of any lost data or password, and more.

3. PROPOSED SOLUTION

In order to achieve all the goals set and stated earlier this paper proposes the introduction of an independent Ethio telecom mobile application, only for smartphones who involve in the E-commerce that Ethio Telecom has setup, which will serve as a secure SMS sending agent that encrypts and sends any SMS or USSD that involves any transfer of money. At the server side the decryption algorithm will be installed. The system uses encrypted messaging protocol with deniability guarantees and message-level forward secrecy. Therefore, no other intruder will be able to read the SMS or access any of the information sent, leaving both sending and receiving parties confident on the transaction. In order to achieve this it will require the usage of Symmetric encryption and identity based encryption.

Unlike, the security solutions mentioned and suggested by other researchers, which require certificate authorities (CA), key exchange sessions, key revocation and generating authorities, and even so they all fail to consider the cost of using multiple SMS on a GSM network for single transaction. The proposed solution uses the advantages we get from using both the symmetric cryptography and Identity based encryption to achieve more robust functionality. Even though, we have several algorithm choices on each encryption type, AES rijandeal is chosen for symmetric encryption and Boneh–Franklin has been selected as our IBE public key encryption. In addition, the proposed solution will use HMAC message authentication function to generate message digest and IBE public key cryptography to digitally sign the message digest. LZW file compression algorithm will also be used to compress the ciphered text incase a message reaches GSM's character limit (i.e. 160 characters).

This solution is novel not only because it achieves all the four constraints, but also it uses phone numbers as a public key and uses a one-time subscription in order to get a corresponding private key eliminating the need for CA (certificate authorities) and key management and distribution issues, replacing all this by a trusted authorities (TA), accessed only by a new user to own a valid key pair. Key exchange session is also removed by appending session key into the ciphered text before generating a message digest. A user has no clue of encryption because it is done behind the user interface of the client application, so it is easy and familiar to use. The basic algorithms are discussed here.

a. Boneh–Franklin ID-PKC

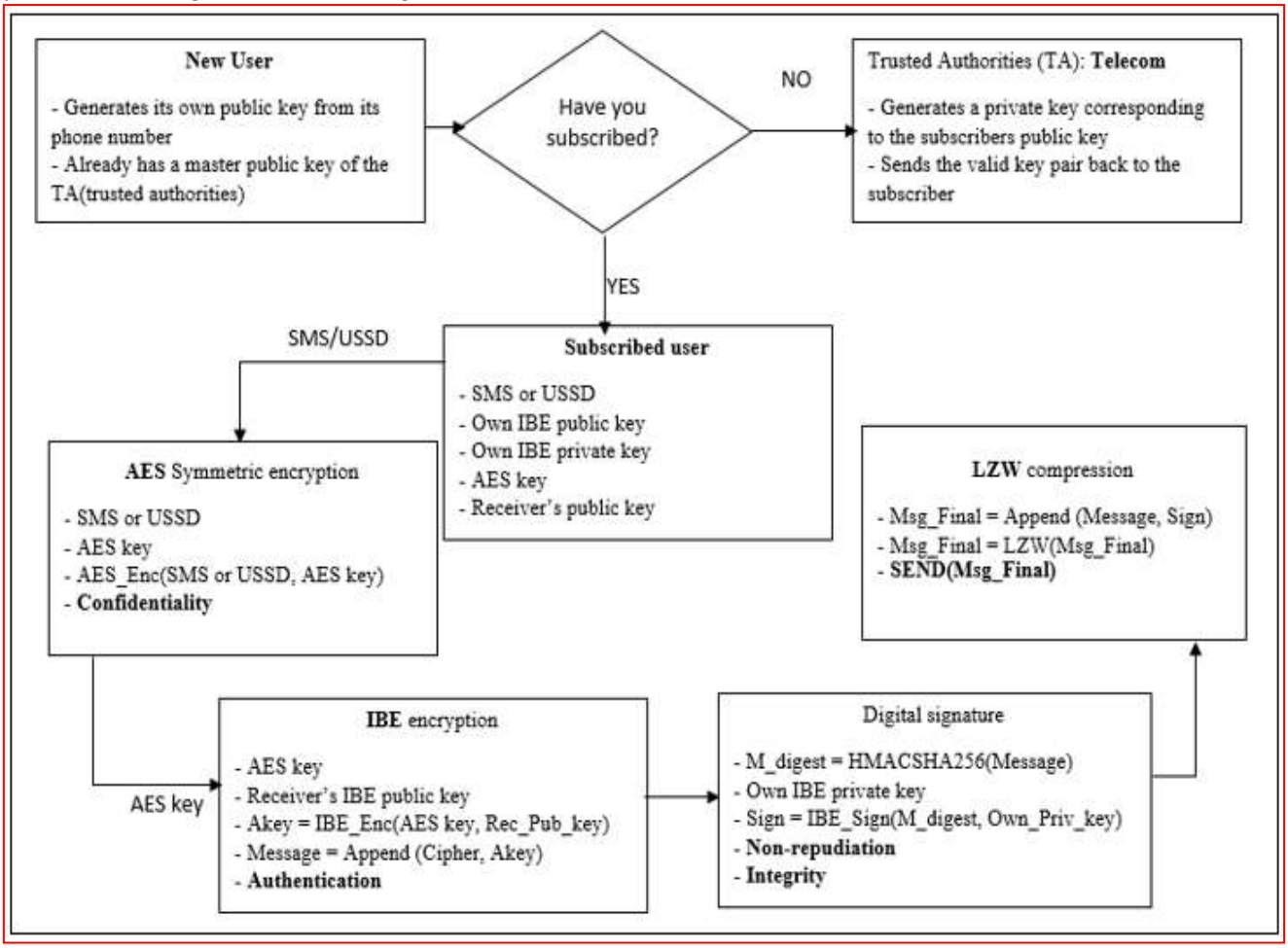
Identity-based systems allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the Private Key Generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as *master key*). Given the master public key, any party can compute a public key corresponding to the identity *ID* by combining the master public key with the identity value. To obtain a corresponding private key, the party authorized to use the identity *ID* contacts the PKG, which uses the master private key to generate the private key for identity *ID*.As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants. This is extremely useful in cases where pre-distribution of authenticated keys is inconvenient or infeasible due to technical restraints. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the PKG [9].

b. AES rijandeal symmetric key cryptograaphy

AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware.[10] Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits and calculations are done in a special finite field.

c. *LZW SMS Compression*

Every SMS's cost is associated with the number of characters it contains. As per the GSM regulation only 160 (English) characters are supported in single message. But sometimes only 160 characters are not sufficient so even if single character exceeds the limit, you will be charged for two messages.



Hence, to solve this problem we apply compression on SMS to pack maximum characters in single SMS body. The LZW algorithm stores strings in a —dictionary" with entries for 4,096 variable length strings. The first 255 entries are used to contain the values for individual bytes, so the actual first string index is 256. As the string is compressed, the dictionary is built up to contain every possible string combination that can be obtained from the message, starting with two characters, then three characters, and so on.

d. *Security protocol design*

As explained briefly in *Fig. 1*.The AES rijndael encryption will use a random number encryption key, generated by AES-Fig. 1. The security protocol design rijndael, to cipher the SMS or USSD and the IBE will use Phone number of users which are unique for everyone and authenticated by Ethio telecom as a public identity specifier to encrypt the AES key and to sign the message digest generated by HMACSHA256 algorithm. Finally, the LZW file compression will compress and send it to the receiver.

4. *IMPLEMENTATION AND TESTING*

Start: ***plaintext*** = input

Step 1: ***ciphertext*** = Private Key cryptography using AES-rijndael to ensure message ***confidentiality***.

- KeyGenerator.getInstance ("DES"), .init (56), and -.generateKey (): Generates the key.
- Cipher.getInstance ("DES/ECB/PKCS5Padding"): Creates the Cipher object (specifying the algorithm, mode, and padding).
- .init (Cipher.ENCRYPT_MODE, key): Initializes the Cipher object,

doFinal(plaintext) : Calculates the ciphertext

Step 2: ciphered key = Public Key cryptography using IBE to transmit the above key.

- Encrypt the symmetric AES key using the public key
- byte [] Encryptkey = (**new** BigInteger(AES_key)) .modPow(Public_key, n).toByteArray();

Step 3: ***mac*** **=**Message authentication code using HMAC-SHA256 ensures message ***integrity***.
- Message = Append ciphertext from step 1 and Encryptkey from step 2
- KeyGenerator.getInstance ("HmacSHA256") and .generateKey (): Generates the key.
- Mac.getInstance ("HmacSHA256"): Creates a MAC object.
- .init (shakey): Initializes the MAC object.
- .update (Message) and .doFinal (): Calculates the MAC object with a plaintext string.

Step 4: signature = Digital signature using IBE-PKC for ***non-repudiation*** purpose.
- Need private key that will only be given by the Trusted Authorities (TA) on the time of subscription
- .initSign (key.getPrivate ()): Initializes the Signature object.
- .update (mac) and .sign (): Calculates the signature
- .initVerify (key.getPublic ()) and .verify (signature): Verifies the signature.

A prototype framework for testing the proposed security solution has been designed as part of the work in this paper. A basic two-tiered model, the client-server model, in which the client requests services in encrypted format and the server decrypts and provides them, is used in the implementation of the security protocol. Application logic is partitioned between client and server. Both of them have software interfaces to run part of the application, establish the connection, and handle the interactions. In addition, the unreliable, error prone, low bandwidth, and high latency wireless networks often require assurances that data has been delivered in a reliable manner. This is accomplished by using intense throwing of the input/output and data exchange exceptions techniques to catch network connection failures.

Technologies Used

Java platform was used due to its Fine-grained control over resource access for both applets and applications and a large number of library functions.
- JSSE (Java Secure Sockets Extension)
- JCE (Java Cryptography Extension)
- Android studio
- Devspace JAVA SMS SDK
- Jetty web socket and server
- Eclipse JavaEE
- Algorithms used: HMAC, AES, LZW, and Boneh–Franklin
- The Sun Microsystems J2ME Wireless Toolkit (WTK) is used on the client side for the application development. The WTK is used to compile, build, package, execute, and as a debugging tool for developing wireless MIDP applications (MIDlets).

The whole project's implementation has two major parts; the first is the client side and the second is the server side.

Client side

This is the side where users see, so it is made easy to use even familiar with the previous user interfaces. The user interface include tabs for SMS sending and USSD dialing. Users have no knowledge of the encryption going underneath. Android studio and eclipse were used to develop the client side application using two main programming languages JAVA and XML. The system is tested and tried on android phones, Samsung to be specific.
- A simple SMS that sends its public key as a means to subscribe
- The server will register the user, calculate appropriate private key and reply it to the sender.
- User will the use these key pair in encryption and signature.

Server side

Since our server side telecom application (TAP) requires actual network provider infrastructure which is impossible to get access, we go through totally different system implementation and testing. JAVA programming language was still used to write the server side decryption-encryption module on eclipse IDE with SMS and USSD libraries and SDKs from Hsenid Mobile Corporation. Devspace simulator and servers form jetty were used to test the telecom application.

Devspace

The Dev Space Telco APIs by HSenid Mobile provide a rich framework for TAP developers to build various server applications by integrating Telco assets such as SMS, USSD, Charging, Location and Subscription to create various types of Telco apps that range from enterprise level to entertainment[10]. To get to our testing we need to do the following.

The application

To run the application you have two options. First one is running as a standalone application. Second one is creating a web archive and deploying in a web container like tomcat.

First option (make sure your codes are on exact places)
- \bin\create_standalone.bat
- cd..\target\stand-alone\bin
- Start-app.bat

Second option
- \bin\create_war.bat

Start Simulator

- Download devspace simulator and extract
- …bin>sdp-simulator.bat console
- http//:localhost:10001 record interface

SMS interface on

- http://localhost:{port of the application}/mo-receiver

USSD interface on

- http://{host}:{port of the sample app}/mo-ussd

According the *table 1.* Below, its efficiency regarding time was no different than the time for the regular SMS and USSD. Despite the tolerable increase in size of overhead that does not lead to violating character limit, it is agreed that, this system is feasible to be implemented and put to use.

Table 1. Time record for the phases in milliseconds

Phases	SAMSUNG-SM-N900A	Devspace simulator
Subscription	6	4
Rijndael (128-bit key)	7	10
IBE encryption	11	9
HMAC	4	4
Signature	12	9

At the end of every security research there is always one important question and that is the performance and cost analysis. GSM, as mentioned earlier, only permits 160 characters for a singular payment i.e. 0.35 cents (ETB). The experiments is performed couple of times to assure that the results are consistent and are valid. It has been established, the application of these algorithms will not lead users to pay twice.

5. Conclusion and Future Work

This application can run on any device which works on Android platform. This application provides a secure, fast, and strong encryption of the data. There is a huge amount of confusion and diffusion of the data during encryption which makes it very difficult for an attacker to interpret the encryption pattern and the plain text form of the encrypted data. The messages encrypted by the developed application are also resistant to Brute Force and pattern attacks. In the future, the application could also provide MMS securing. It could be ported to other programming platform for greater distribution among users and also. Most importantly, an independent key management or certificate authorities could be implemented in order to use several public key cryptographies.

Acknowledgment

This research was supported by a friend, brother and mentor Mr. Hiliwi Leake kidane, who I want to thank very greatly. My gratefulness thanks to Ramesh Gadde, M.Tech (Ph.D) and Kifle Berhane. N - M.Tech, my project advisor who was committed to the end helping me in every ways possible. I finally thank my roommates and friends who provided insight and expertise that greatly assisted the paper.

References

[1] Hello Cash, services, http://hellocash.et/personal/how-it-works.

[2] all Africa, Ethio-telecom, Office sign 8100 A SMS service accord 26 Sep 2015 National News *By* Haftu Gebrezgabiher.

[3] http://lehulu.kifiya.com/, service promises.

[4] Security in the GSM Network by Ammar Yasir Korkusuz, Bogazici University, Electrical-Electronics Engineering Department, 2015.

[5] R.Rayarikar,S. Upadhyay and P. Pimpale,"SMS Encryption using AES Algorithm on Android".

[6] A novel peer-to-peer SMS security solution using ahybrid technique of NTRU.

[7] Hassinen M, Markovski S (2003). Secure SMS messaging using Quasigroup encryption and Java SMS API. In: SPLST'03, Finland.

[8] SANS Institute InfoSec Reading Room, PGP: A Hybrid Solution by Jessica J. Benz, GIAC certification version 1.2e, 2015.

[9] Public-Key Cryptography -- PKC 2015: 18th IACR International conference on practice and theory in public key cryptography.

[10]http://wso2experience.blogspot.com/2014/05/how-to-run-ussd-sms-lbs-devspace.html.

[11] A Security Protocol of Mobile E-Commerce Based on SMS by Liujian, 978-1-4244-8625-0/11/$26.00 ©2011 IEEE.

[12] Data security using private key encryption system based on arithmetic coding, Ajit Singh1 and Rimple Gilhotra2 2013.

[13] Forward-Security in Private-Key Cryptography, Mihir Bellare∗ Bennet Yee† *2012.*

[14] Securing SMS using Cryptography, Sri Rangarajan, N. Sai Ram, N. Vamshi Krishna, IJCSIT 2013.

[15] Balageru Mirt Finalists and Voting Information. http://www.zeethiop.com/watch.php?vid=fdfebcb96.