

EVALUATION OF EESP PERFORMANCE IN COMPARISON TO CURRENT SECURITY PROTOCOL

Omprakash Maru^{1*}, Dr. Ramesh Vishwakarma²

¹Research Scholar

²Guide, Rabindranath Tagore University, Bhopal.

*Corresponding Author

Omprakash Maru

Research Scholar.

Article History

Received: 08.09.2024

Accepted: 19.09.2024

Published: 17.10.2024

Abstract: It is normal practice to send and receive short messaging service (SMS) messages using a mobile phone. As a result, mobile phones are indispensable for the communication of information in a secure and instantaneous manner in our everyday lives. While utilizing mobile computing, this is one of the most efficient techniques for exchanging data in a safe manner. These days, researchers are concentrating their efforts on mobile computing as a technology that can send data in a safe and effective manner. Ensuring that end users are able to communicate securely with one another from beginning to finish is essential for protecting against it. We propose a method that encrypts SMS data from the beginning to the end as it travels over the network in order to accomplish the above-and-beyond. In addition to being an effective and safe protocol, EasySMS makes it possible for end-to-end encrypted SMS communication to take place between end users. The implementation of this protocol is the objective of this endeavor. Two mobile stations, the authentication server (AS) and the relevant mobile station (MS), are responsible for storing all of the symmetric keys that are shared between the AS and the MS. The registration authority is responsible for storing all of the information that is associated with the mobile subscriber. Simply said, these are the primary elements that make up the Easy SMS system. The only thing that happens during the first step is authentication, but during the second stage, the two mobile phones really communicate with one another in order to send and receive messages before the time limit is approached. Two different scenarios are included in the Easy SMS protocol. In the first scenario, both mobile stations are connected to an authentication server (AS), which is required for authentication. When it comes to the second scenario, the mobile stations make use of two distinct authentication servers (AS). A number of different encryption methods are used while transmitting data over the short message service (SMS). These methods include the Advanced Encryption Standard (AES), Triple DES, and the MD5 algorithm, which provides a hash function. The Easy SMS protocol offers protection against a variety of threats, including impersonation, replay, man-in-the-middle, over-the-air modification, and SMS disclosure, to name a few.

Keywords: SMS, protocol, EESP, network.

Cite this article:

Maru, O., Vishwakarma, R., (2024). EVALUATION OF EESP PERFORMANCE IN COMPARISON TO CURRENT SECURITY PROTOCOL. *ISAR Journal of Science and Technology*, 2(10), 57-63.

Introduction

Encryption is a solution that can be relied upon when it comes to effectively securing sensitive information. Understanding the secret key is necessary in order to decrypt a file that has been encrypted. The term "plain text" is used to represent data that has not been encrypted, while the term "cipher text" depicts data that has been encrypted. An algorithm that generates an encryption key is one of the components that make up the encryption technology. For a system of encryption that has been well devised, it is necessary to have a large amount of computational expertise and resources in order to decode the message without the key. Only authorized users are able to decode the encrypted message using the key that is provided by the sender to the recipient.

Network encryption, sometimes known as network-level encryption, is a method of networking security that offers encryption services at the network transfer layer. This method is also often referred to as network-level encryption. Encryption of networks makes use of the Internet Security Protocol (ISP) and other protocols that are connected to it in order to improve and broaden the scope of online privacy.

The use of ciphers that cannot be broken without the decryption key has lately been a heated matter of discussion with many people. Encryption is seen by some companies as a realistic method of preventing unauthorized access; nevertheless, the great majority of enterprises see it as a method of securing sensitive data and restricting the ability of interceptors to gain it.

As a result of advancements in data security over the last several

years, secure socket layer cryptography has gained popularity as a method for users to encrypt sensitive data while it is being sent over the internet. Because of this, the subject of cyber security has become an increasingly urgent one. An investigation that was conducted not too long ago by a firewall manufacturer states that Secure Socket Layer encryption examines both incoming and outgoing signals in order to conceal the content of the information that is being sent. On the other hand, with the exception of those who possess the ability to decode data, nearly no one is able to ascertain the essence of the information that is sent over the network. The development of surveillance manifestations has not been able to keep up with the doubling of the amount of network traffic that occurs at the secure socket layer.

Shakeeba S. Khan and Professor R.R. Tuteja (2015) discussed both the symmetric Data Encryption Standard (DES) and the asymmetric cryptographic technique (RSA). Both of these encryption methods are used to protect sensitive information. The next step that they took was to provide specifics on how to address the difficulties associated with cloud storage security by integrating two distinct technologies, such as DES and RSA.

Review of the Literature

In order to make the data that is kept on the cloud more secure, Jasleen Kaur and Dr. Sushil Garg (2015) advised employing a mix of techniques for encryption and decryption. These algorithms include RSA, RSA Digital Signature, Fiestel, and XOR operations.

authored by Eveque Mutabaruka (2016) The hybrid encryption approach that was proposed by the researchers takes use of both symmetric and asymmetric encryption processes. These processes include Rivest Shamir Adleman and the enhanced Encryption Standard. The fact that it used symmetric key encryption analysis followed by asymmetric key distribution was a significant contributor to the success of this endeavor. In order to make it much more secure, it is important to investigate the possible measures that may be taken to protect both the data and the key by using hybrid encryption. We make use of practical approaches such as digital signatures, hashing functions, symmetric algorithms such as AES and RSA, and asymmetric algorithms such as RSA in order to make the data more secure. This improvement is based on cryptography since it is capable of meeting the standards established for security. In addition, we examine the features, consider the similarities and differences between the two approaches, and discuss the safety precautions. In the event that we have finished debating it, we shall proceed to design a hybrid encryption system.

Researchers Shikha Rani and Shanky Rani (2016) compiled a collection of publications that were pertinent to the topic of cloud data security after reviewing past research on the subject. For the purpose of enhancing the server's safeguards when it is hosted in the cloud, they suggested a hybrid security encryption approach that used both Blowfish and MD5.

In 2016, Vinita Keer, Dr. Syed Imran Ali, and Neeraj Sharma conducted research on a number of well-known cloud service companies. Google Drive, Microsoft Azure, and One Drive were among the providers that were included in this group. Some of the cryptographic methods that are most often used in cloud computing were among those that were investigated. Current cryptography includes examples of algorithms like AES, RC6, DES, 3DES, and

BLOWFISH, which are all instances of cryptographic algorithms. The searchable and homomorphic encryption algorithms, in addition to the attribute-based encryption method, are also included in this category of algorithms. In addition to this, they came up with the concept of hybrid encryption, which is the combining of two or more cryptographic techniques in order to safeguard data that is stored in the cloud by using the benefits that each approach offers.

Xu, Yonglin, Wu, Shaofei, Wang, Mingqing, and Zou, Yuntao all contributed to the publication of a work in the year 2020. A significant increase in the amount of medical data has occurred as a result of the introduction of big data and the extensive incorporation of the Internet into the healthcare industry. Identifying a method that could guarantee the safety of the data was of the utmost importance. Traditional techniques of encryption, on the other hand, were not able to keep up with the processing needs of enormous data collections. The purpose of this study is to create and investigate a distributed RSA encryption method. The technique is based on a distributed field system and programming paradigm. The article is built on the open-source Hadoop project, which acts as its basis. This article starts out with a suggestion for the design strategy that should be taken for a platform-based distributed encryption method. Additionally, the algorithm's distribution is formed, process management and module partitioning are carried out to the fullest extent of their capabilities, and the distributed encryption function of the single computer encryption system is accomplished. Lastly, the approach of distributed encryption is put through its paces in an experimental setting by using the large-scale distributed cluster that was constructed in the laboratory. It is in this cluster that the effectiveness, adaptability, and usefulness of the strategy are put to the test. There is evidence to imply that the distributed encryption technique may have been used in order to rapidly process large amounts of data and manage vast amounts of data.

In 2021, Abdullah Mahmood said that this study investigates cryptography as a means of securing sensitive data during network transmission. This research is being conducted for a variety of reasons, including the fact that it is being conducted between individual users, enterprises, organizations, or cloud services. In order to prepare the data for transmission to the receiver on the network, it is necessary to first encrypt it using the cryptographic technique. Decryption was the second step, which required the receiver to demonstrate that they were the owner of the original data. In order to compare and contrast three different encryption methods—AES, RSA, and a hybrid algorithm—this study looked at how long each method required to decode data. The goal of this research was to compare and contrast these three methods. Following the completion of the investigation, it was determined that the hybrid algorithm offered the best degree of security in comparison to the other techniques.

Method: Security Goals and Proposed Solution

Attack Model

Within the framework of the attack model, we will investigate a wide range of attack scenarios, examining the myriad of ways in which a malicious mobile station (MS) might get access to authentication information and even fool a legitimate one. Since the contents of a short message service (SMS) message are sent as

plain text to the short message service center (SMSC), it is common knowledge that the person in charge of the network may easily see the contents of the message.

An encryption technique that is very weak, such as A5/1 or A5/2, has been used in order to ensure the safety of the over-the-air interface that exists between the mobile station (MS) and the base station transceiver (BTS). With this knowledge, it is possible for an adversary to simply read the algorithm information that is included in an SMS message, or they may modify the SMS information and transmit it. An further attack vector comes in the form of an effort by the attacker to generate the keys for the authentication protocol. As was described in the previous message that was followed by the replay attack, the attacker has the ability to try to delay the conversation that is taking place between mobile stations (MS1) and (MS2), and it also has the ability to collect vital data that is being sent throughout the execution of the protocol. In order to do authentication, it may either provide the server the information that it is aware of or it may modify the information that is sent in a certain sequence. When a mobile station (MS) is connected to a base station (BTS) over a wireless network, there is also the possibility of a man-in-the-middle assault from taking place. An end will be brought to the session that was generated by the mobile station (MS) that has been verified. This will result in the assailant forming a new connection with the victim, as well as the victim as well. On the active connection, it will monitor the session that is taking place. Additionally, it is able to intercept and modify messages that are being sent. There is a possibility that an unauthorized individual may introduce false information or intercept communications that are being sent between mobile stations (MS).

The transmission of sensitive information is considered to be compromised if an unexpected format or an insufficient encryption mechanism is involved. As soon as an adversary has the secret key or even a clue about it, all of the aforementioned possibilities become conceivable. It is likely that it will also make an effort to learn the session key if it is feasible to carry out this sort of attack during the phase in which users exchange keys. In the event that the right integrity is not maintained, it is possible for the mobile station (MS) and the authentication server (AS) to be impersonated via the transmission of data across the network. It is possible for an attacker to assume the identity of a legitimate MS in order to deceive the AS into believing that they are seeking authentication tokens from a genuine MS. It is also possible for him or her to assume the identity of a valid AS and then request that the legitimate MS deliver the information. This would cause the target MS to believe that the information originated from a genuine AS.

Problem Definition

There is no facet of our life that does not include the use of short message service (SMS). Among the many applications that may be carried out using short message service (SMS), some examples include the transmission of financial information, license numbers, passcodes, and other information. On the other hand, some mobile service providers do not offer a high level of security when it comes to delivering short messaging service (SMS) messages via the networking system. Easy SMS, a protocol that is both efficient and secure, has been used in order to handle the security problem. This protocol offers a service that ensures the safe transmission of SMS from beginning to finish. Through the use of a variety of

encryption techniques, we are able to circumvent a wide range of threats.

Analysis: Proposed Protocol

The brand-new protocol known as EasySMS provides end-to-end secure data transport over cellular networks in two different scenarios. As can be seen in Figure 3.1, the first scenario involves two MS that are both members of the same AS and, as a result, make use of the same Home Location Register (HLR). Under the second scenario, which is shown in Figure 3.2, the two MS are members of different AS and, as a result, make use of different HLRs. There are two key components that make up the EasySMS method of communication. The first thing that has to be done is for each MS to connect with the Authentication Server (AS), which also serves as the Authentication Center (AuC) and maintains a record of all of the symmetric keys that are shared between the two. The acronym AS will be used to refer to AuC from this point on. On the other hand, the Registration Authority (RA) is another agency that is responsible for monitoring the data that pertains to mobile users.

The abbreviations and symbols are:

MS: Mobile station referring user

AS: Authentication server referring AuC

RA: Registration authority

IDMS: International mobile subscriber identity of MS(128-bits)

Q: New session identifier(28-bits)

Rc/Nc/Na/Ns: Random Number(128-bits)

Pf: Private Port Number(16-bits)

ReqNo: Request Number(8-bits)

SK/SK_MS: Symmetric key shared between MS and AS(128-bits)

DK1: Delegation key(256-bits)

MAC/H: Message Authentication Code/Hash(64-bits)

SQ: sequence number(28-bits)

Ti:Timestamp(64-bits)

CertSAG: Certificate of SecurityAccessGateway(40- bits)

PK: public key of the server (128-bits)

UAKey: Primary Key(128-bits)

Expiry/ExpT: Expiry Time(64-bits)

SK_AS-CA: Symmetric key shared b/w AS and CA/RA(128-bits)

SK_AS1-AS2: Symmetric key shared b/w AS1 and AS2(128-bits)

F1: Message Authentication Code Function

Scenario-1: Both MS Belong to Same AS

The scenario shown in Figure 1 is one in which two message servers, MS1 and MS2, are both members of the same AS and interact with one another. The problem may be broken down into two distinct parts.

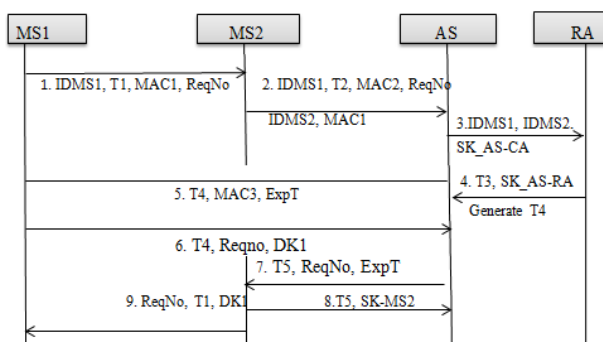
Initial phase: (1) The mobile user (MS1) must first begin the connection with another mobile user (MS2) in order to receive an

SMS. This is the first stage in the SMS sending process. A timestamp T_1 , a request number $ReqNo$, and a message authentication code $MAC1 = f1SK1 (IDMS1 || ReqNo)$ are the remaining components of this request. The first component of this request is the International Mobile Subscriber Identity (IMSI) of MS1 (for example, IDMS1). In this particular instance, the MS1 and the AS2 both have the same symmetric key, which is denoted by the letter SK1. When MS1 delivers a message to another mobile user, which we will refer to as MS2, the second user computes the $MAC2$ by using the formula $f1SK2 (IDMS2 || T2 || MAC1)$ to the message. The next step is for MS2 to interact with the AS by means of a message that contains the following information: IDMS1, IDMS2, T_2 , $MAC1$, $ReqNo$, and $MAC2$. Within this context, IDMS2 refers to the MS2 IMSI. There is a symmetric key, SK2, that is shared by both the MS2 and the AS. The goal of this message is for MS2 to inquire with the AS about the legitimacy of IDMS1 and to request that they validate it. Following the receipt of a message from the MS2, the AS compares the $MAC2$ that was received with the $MAC2'$ that was calculated using the formula $f1SK2 (IDMS2 || T2 || MAC1)$. Using a symmetric shared key (SK_AS-CA) between the AS and CA/RA, the AS sends IDMS1 and IDMS2 to the CA/RA together with a timestamp T_3 if it is valid. This is done in order to verify the identities of both MS. If the distance between $MAC2$ and $MAC2'$ is not equal, the connection will be terminated. After then, the CA/RA is responsible for confirming the validity of both entities by using the timestamp T_3 , and then it sends the answer back to the AS. If the AS judges that any of the entities are invalid after getting the message from the CA/RA, the connection is immediately broken, and MS1 is forced to submit a new connection request. This occurs immediately after the message is received. In the event that both entities are genuine, the AS will generate a new timestamp T_4 , an expiration time to authenticate MS1 (let's say $ExpT$), a delegate key DK1 from the SK1 by using a function f_2 , and a new message authentication code $MAC3 = f1SK1 (T4 || ExpT || ReqNo)$ and $DK1 = f2SK1 (T4 || ReqNo)$. After that, the AS will transmit T_4 , $MAC3$, and $ExpT$ to the MS1 in order to do so. The MS1 immediately begins the process of calculating $MAC3'$ and comparing it to the $MAC3$ that it has received as soon as it gets the message from AS. ($T4 || ExpT || ReqNo$) $MAC3'$ was equal to $f1SK1$. In the event that they are identical, MS1 will compute DK1. MS1 then sends T_4 and

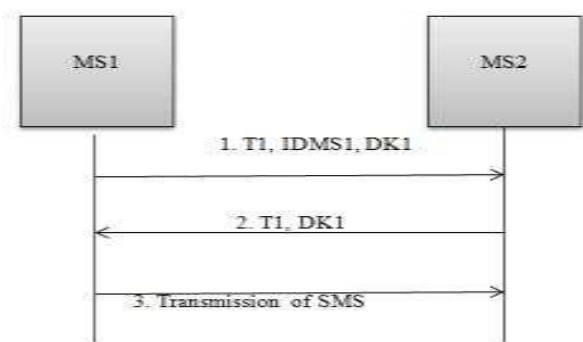
the matching $ReqNo$ to the AS by using the DK1 key in the transmission processes. By comparing the received T_4 to the value that it has cached, the AS is able to verify the $ReqNo$. In the event that both are correct, the authentication of MS1 is considered complete. Once all of the data has been encrypted using the MS2 SK (SK_MS2), which is a key that is shared between the two entities, the AS will then send DK1 to MS2 along with revised timestamps T_5 , $ExpT$, and $ReqNo$. Through the process of responding to the AS, the MS2 verifies that it has received the DK1 key. The T_5 is then encrypted by utilizing the SK of the MS2. For the purpose of MS1 confirming that T_1 and $ReqNo$ are correct, MS2 also sends them to MS1 encrypted using DK1 in order to convey the information. Furthermore, this message serves as confirmation that the MS2 in question has properly received DK1.

The second phase involves the exchange of a secret symmetric key between the two MS. This allows for the secure transmission of messages via the use of a cryptographic method that is both suitable and resilient, such as AES or MAES. After phase one is finished, a session is formed that enables the two MS to interact in a safe manner for a certain period of time, which is denoted by the acronym $ExpT$. The DK1 key is used for ciphering purposes by both MS1 and MS2 throughout this period of time. However, after the $ExpT$ time has passed, the session will be terminated, and MS1 will be required to send a new request to MS2 using the same protocol as phase-1 and a new request number, which will be denoted by $ReqNo$. The following processes are carried out in order to ensure that the two MS are able to communicate with one another inside the $ExpT$:

The timestamp, which we will refer to as T_i , and the IDMS1 are both encrypted by the MS1 using the symmetric key (DK1) of the MS1, and then they are sent to the MS2 for further processing. In order to verify the authenticity of IDMS1 and determine whether or not T_i is less than $ExpT$, MS2 decrypts the message by using the same DK1 key. Upon validation of these prerequisites, the verification of MS1's identity and validity as a connection user is considered to be complete. Following then, MS2 confirms that it has received the message from MS1 by sending back the identical T_i that has been encrypted using DK1. A encrypted SMS discussion is taking place between both MS.



(a)



(b)

Figure 1: Phase 1 and Phase 2 of the EasySMS Protocol (Vaidehi, 2016)

Scenario-2: Both MS Belong to Different AS:

Figure 2 illustrates a scenario in which two message servers, MS1 and MS2, belonging to different ASs connect with one another and interact with one another. Within the context of this scenario, the two mobile users are located in different authentication centers and are located in different areas of the continent. In the event that two MSs come from different service providers, it is possible for them to employ distinct

authentication departments. In a similar manner, this circumstance may be divided into two halves.

Phase-1: The information is presented in the same manner as the first step of scenario 1. A symmetric key known as SK1 is shared by MS1 and AS1 in this particular instance. IDMS1, IDMS2, ReqNo, T2, MAC1, and MAC2 are the pieces of information that are sent from the MS2 to the AS that it is associated to since we will refer to it as AS2. There is a symmetric key, SK2, that is shared by both the MS2 and the AS2. Within the context of this communication, the MS2 requests that the AS2 confirm that the IDMS1 is a genuine entity. The timestamp T1 is saved in memory by the MS2 once it has been received from the MS1 and received. The identical computation that was performed in scenario-1 is carried out by the AS2 in order to ascertain if $MAC2 = MAC2'.CA/RA$ sends a reply back to AS2 after confirming the legitimacy of both entities, using the date T3 and the identifier of the AS to which MS1 belongs (let's say AS1). This is done after the CA/RA has confirmed that both entities are genuine. Following the AS2's verification that both entities are genuine, it will either make use of a secure channel or a symmetric key that is shared by both systems, such as SK_AS1-AS2, in order to send (IDMS1, ReqNo, MAC1) to the AS1. The technique described here is the same as step 5 of scenario 1. That all AS communicate with one another by using the symmetric shard keys that have previously been produced is something that we consider as a given. When it receives the message from the AS2, the AS1 does the calculation $MAC1' = f1SK1 (IDMS1 || ReqNo)$ before continuing with the comparison of the two. In the event that the two are not compatible, we sever our connections. Assuming that they are the same, the AS1 will generate a new timestamp T4, an expiry time for authenticating MS1 (which we will refer to as ExpT), a delegate key DK1 that is generated from MS1's SK1 via the use of a function f2, and a message digest key MAC3. MAC3 is represented by the formula $f1SK1 (T4 || ExpT || ReqNo)$, whereas DK1 is represented by the formula $f2SK1 (T4 || ReqNo)$. After then, AS1 will transmit (T4, MAC3, and ExpT) to MS1 in order to connect with Microsoft. Exactly as in scenario-1, MS1 will send (T4, ReqNo) to AS1 encrypted with the DK1 key after AS1 has transmitted a message (as described in the previous sentence). The AS1 checks the T4 and the ReqNo once scenario-1 has been completed. First, AS1 uses the SK_AS1-AS2 key to verify MS1's authentication by connecting with AS2 via a message (ReqNo, ExpT, and DK1) in order to certify MS1's authenticity. The AS2 encrypts DK1 by using the shared key between the two devices, then adds a new timestamp T5, an expiry time ExpT, and a request number ReqNo, and finally sends it to the MS2 for transmission. The procedure has 10 MS2 sending the encrypted answer of T5 to the AS2, exactly as it did in step 8 of scenario 1. As was the case in scenario 1, step 9, it is quite similar.

Phase-2: The two MSs may safely transmit message data using an appropriate method once they have a shared secret symmetric key.

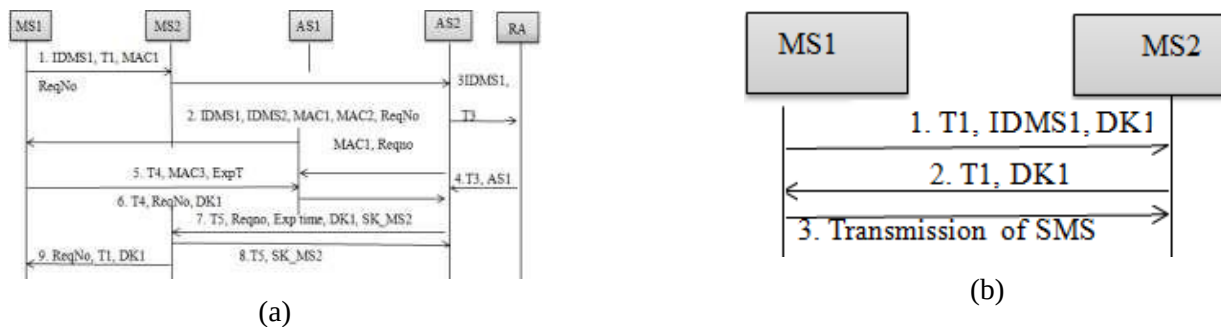


Figure 2: Scenario 2 for the EasySMS Protocol: (a) Phase-1; (b) Phase-2

There are a number of cryptographic algorithms, including AES and MAES, which will be addressed in subsequent sections in further depth. Phase 1 comes to a close with the establishment of a session that, for the length of ExpT, enables the two MS to interact with one another in a safe manner. The DK1 key is used for ciphering purposes by both MS1 and MS2 throughout this period of time. However, after the ExpT time has passed, the session will be terminated, and MS1 will be required to send a new request to MS2 using the same protocol as phase-1 and a new request number, which will be denoted by ReqNo. The following processes are carried out in order to ensure that the two MS are able to communicate with one another inside the ExpT:

The timestamp, which we will refer to as Ti, and the IDMS1 are both encrypted by the MS1 using the symmetric key (DK1) of the MS1, and then they are sent to the MS2 for further processing. In order to verify the authenticity of IDMS1 and determine whether or not Ti is less than ExpT, MS2 decrypts the message by using the same DK1 key. Upon validation of these prerequisites, the verification of MS1's identity and validity as a connection user is considered to be complete. Following then, MS2 confirms that it has received the message from MS1 by sending back the identical Ti that has been encrypted using DK1. A encrypted SMS

discussion is taking place between both MS.

Security Analysis with Confidence Interval

In addition, we have estimated the range of the confidence interval for each technique that accepts 160 characters as input, taking into consideration 95% of the total. Typically, the stated margin of error is almost twice as large as the standard deviation. This is the reason behind this. The reliability of an estimate may be shown by the use of a confidence interval, which is an estimate of a population parameter that is based on interval calculation. It reflects the range of confidence intervals (high and low range values) for both the encryption and decryption of the message (SMS) with 160, 320, 480, 640, and 800 characters in length. This is the case for the DES algorithm, Triple-DES with two keys, Triple-DES with three keys, and the Advanced Encryption Standard (AES) algorithm. In nanoseconds, the timestamps are shown. For the purpose of determining the confidence interval, we have used the t-distribution. This is due to the fact that, in the event that the data does not adhere to a normal distribution, the t-distribution is able to compute confidence intervals for large n', which in our instance is 100 samples. A text message (SMS) may contain anywhere from 160 to 800 characters, and this technique determines how many

characters it can have. An SMS that contains more than 160 characters is split up and combined with another SMS if it is longer than that. As a consequence of this, the amount of bits that make up a message that has been sent may range anywhere from 11,120 bits to 5,6000 bits, with a seven-bit ASCII value being given to each letter. In situations where the standard deviation is low, the data points are more closely packed around the mean, while in situations where the standard deviation is large, the data points are more widely spread. Because it follows so rigidly to its output range, the Advanced Encryption Standard (AES) algorithm is better.

Bandwith Utilization

We will compare and contrast the three protocols in this section depending on the amount of bandwidth that each of them requires. This article examines the bandwidth usage of EasySMS Protocol in comparison to that of SMSsec and PK-SIM. Whether the number of authentication requests is 10, 50, 100, 200, 500, or 1000, respectively, the average bandwidth use during authentication is 51% lower with the EasySMS Protocol than with SMSsec and 31% lower with PK-SIM. This is the case regardless of the magnitude of the number of authentication requests. Similar to how EasySMS reduces the quantity of messages sent and received by 62% and 45%, respectively, when compared to both protocols, EasySMS is considered to be superior.

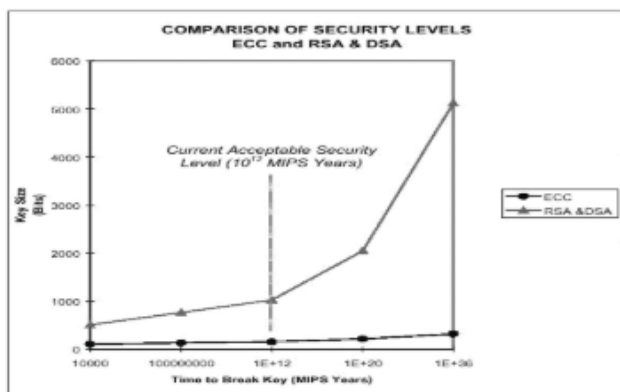


Figure 3: Comparison of AES, DES, and Regular AES in terms of their temporal complexity

Table 1: Calculation of Execution time

t_aes (ms)	t_des(ms)	t_stdaes(ms)
102866	59684	111633
150753	72354	113930
160376	70766	120510
75279	52055	81021

AES's Average Time: 122318.5 ms DES's Average Time: 63714.75 ms Standard time of arrival: 106773.5ms

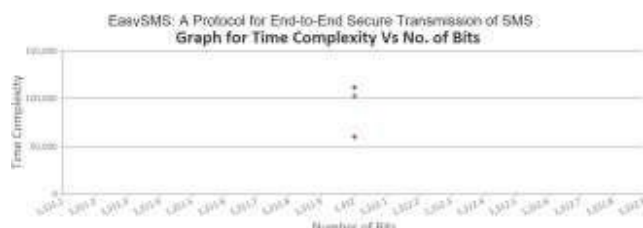


Figure 4: Time Complexity vs Bit Count Graph

Conclusion

This effective protocol makes it possible to send and receive messages using the secure messaging service (SMS) from beginning to finish. In addition to maximizing the use of available bandwidth, this protocol reduces the total number of messages that are sent. There are two steps included in the first level. When the Easy SMS protocol is in its authentication phase, the mobile stations check each other's identities to ensure that they are who they claim to be. It is possible that the actual message exchange between the stations will take place after this has been completed successfully. The procedures of encryption and decryption include participation from both the transmitter and the recipient of the message. The database of the Authentication Server (AS) is where the symmetric keys are stored carefully and securely. In comparison to asymmetric algorithms, symmetric algorithms are favored due to the fact that they are much faster. When it comes to the execution time, the AES method takes 102.866 seconds, whereas the DES algorithm takes 59.684 seconds. Using the DES technique for encryption is a speedy process. The technique might be improved in a variety of ways in later endeavors, and there are so many possibilities. Real-time implementation of the Protocol is a possibility under certain circumstances. Because it is scheduled to be implemented over the course of the next several years, it is possible that malware assaults may be reduced. The need of securing the highest possible level of security cannot be overstated in light of the ever-increasing capabilities of SMS technology. By doing so, users will be able to send and receive private messages, authenticate their identities, and reduce the likelihood of information being discovered by unauthorized parties. Considering how strong and tough it is to breach, this protocol has the ability to stop attacks that are freshly linked.

References

1. Press Release. (2012, Dec. 3). Ericsson Celebrates 20 Years of SMS.
2. Anderson, R. E., Brunette, W., Johnson, E., Lustig, C., Poon, A., Putnam, C., ... & Borriello, G. (2010, December). Experiences with a transportation information system that uses only GPS and SMS. In *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development* (pp. 1-10).
3. Risi, D., & Teófilo, M. (2009, September). MobileDeck: Turning SMS into a rich user experience. In *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems* (pp. 1-4).
4. Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., & Naik, V. (2011, March). SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications* (pp. 1-6).
5. Chen, J., Subramanian, L., & Brewer, E. (2010, September). SMS-based web search for low-end mobile devices. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (pp. 125-136).
6. DeRenzi, B., Findlater, L., Payne, J., Birnbaum, B., Mangilima, J., Parikh, T., ... & Lesh, N. (2012, March).

- Improving community health worker performance through automated SMS. In *Proceedings of the fifth international conference on information and communication technologies and development* (pp. 25-34).
7. Densmore, M. (2012). Experiences with bulk SMS for health financing in Uganda. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 383-398).
8. Hellström, J., & Karefelt, A. (2012, March). Participation through mobile phones: A study of SMS use during the Ugandan general elections 2011. In *Proceedings of the fifth international conference on information and communication technologies and development* (pp. 249-258).
9. Murynets, I., & Piqueras Jover, R. (2012, November). Crime scene investigation: SMS spam data analysis. In *Proceedings of the 2012 Internet Measurement Conference* (pp. 441-452).
10. Park, K., Ma, G. I., Yi, J. H., Cho, Y., Cho, S., & Park, S. (2011, January). Smartphone remote lock and wipe system with integrity checking of SMS notification. In *2011 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 263-264). IEEE.
11. Nehra, A., Meena, R., Sohu, D., & Rishi, O. P. (2012, March). A robust approach to prevent software piracy. In *2012 Students Conference on Engineering and Systems* (pp. 1-3). IEEE.
12. Gligoric, N., Dimcic, T., Drajić, D., Krco, S., & Chu, N. (2012, November). Application-layer security mechanism for M2M communication over SMS. In *2012 20th Telecommunications Forum (TELFOR)* (pp. 5-8). IEEE.
13. Gupta, S., Sengupta, S., Bhattacharyya, M., Chattrejee, S., & Sharma, B. S. (2009, November). Cellular phone based web authentication system using 3-D encryption technique under stochastic framework. In *2009 First Asian Himalayas International Conference on Internet* (pp. 1-5). IEEE.
14. Traynor, P., Enck, W., McDaniel, P., & La Porta, T. (2006, September). Mitigating attacks on open functionality in SMS-capable cellular networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking* (pp. 182-193).
15. Zeng, Y., Shin, K. G., & Hu, X. (2012, April). Design of SMS commanded-and-controlled and P2P-structured mobile botnets. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks* (pp. 137-148).
16. Hamandi, K., Elhajj, I. H., Chehab, A., & Kayssi, A. (2012, October). Android SMS botnet: a new perspective. In *Proceedings of the 10th ACM international symposium on Mobility management and wireless access* (pp. 125-130).
17. Lo, J. L. C., Bishop, J., & Eloff, J. H. (2008). SMSec: An end-to-end protocol for secure SMS. *Computers & Security*, 27(5-6), 154-167.
18. Rongyu, H., Guolei, Z., Chaowen, C., Hui, X., Xi, Q., & Zheng, Q. (2009). A PK-SIM card based end-to-end security framework for SMS. *Computer Standards & Interfaces*, 31(4), 629-641.
19. Hassinen, M. (2006, April). Java based public key infrastructure for sms messaging. In *2006 2nd International Conference on Information & Communication Technologies* (Vol. 1, pp. 88-93). IEEE.
20. Wu, S., & Tan, C. (2009, October). A high security framework for SMS. In *2009 2nd International Conference on Biomedical Engineering and Informatics* (pp. 1-6). IEEE.
21. De Santis, A., Castiglione, A., Cattaneo, G., Cembalo, M., Petagna, F., & Petrillo, U. F. (2010, February). An extensible framework for efficient secure SMS. In *2010 International Conference on Complex, Intelligent and Software Intensive Systems* (pp. 843-850). IEEE.
22. Toorani, M., & Beheshti, A. (2008, July). SSMS-A secure SMS messaging protocol for the m-payment systems. In *2008 IEEE Symposium on Computers and Communications* (pp. 700-705). IEEE.
23. Mondal, P., Desai, P., Ghosh, S. K., & Mukherjee, J. (2013, January). An efficient SMS-based framework for public health surveillance. In *2013 IEEE Point-of-Care Healthcare Technologies (PHT)* (pp. 244-247). IEEE.
24. Yang, C. C., Tang, Y. L., Wang, R. C., & Yang, H. W. (2005). A secure and efficient authentication protocol for anonymous channel in wireless communications. *Applied Mathematics and Computation*, 169(2), 1431-1439.
25. Khiabani, Y. S., Wei, S., Yuan, J., & Wang, J. (2012). Enhancement of secrecy of block ciphered systems by deliberate noise. *IEEE Transactions on Information Forensics and Security*, 7(5), 1604-1613.
26. Wei, S., Wang, J., Yin, R., & Yuan, J. (2013). Trade-off between security and performance in block ciphered systems with erroneous ciphertexts. *IEEE Transactions on Information Forensics and Security*, 8(4), 636-645.
27. Biham, E. (1998). Design tradeoffs of the AES candidates. *Lecture Notes in Computer Science*. New York, NY, USA: Springer-Verlag.
28. Ray Rischpater. (2009). Beginning Java™ ME Platform. Messaging with Wireless API, part 4, pp. 373-407.
29. E. Biham, A. Shamir. (1991). Differential Cryptanalysis of DES Cryptosystems. *J. of Crypto.*, 4(1), pp. 3-72.
30. J. Choi I, J. Kim, J. Sung, S. Lee, J. Lim, "Related-Key and Meet-in-the-Middle Attacks on Triple-DES and DES-EXE," ICCSA, LNCS 3481, 2005, pp. 567-576.
31. D. Altman, D. Machin, T. Bryant. (2000). Statistics with Confidence Intervals and Statistical Guidelines. BMJ Books, 254 pages.
32. W. N. Venables, B.D. Ripley. (2002). Modern Applied Statistics with S. Springer, 4th Edition, XI, 497 pages.