Public Key Encryption with Keyword Search

Dan Boneh[1]*, Giovanni Di Crescenzo[2], Rafail Ostrovsky[3]**, and Giuseppe Persiano[4]***

[1] Stanford University
dabo@cs.stanford.edu
[2] Telcordia
giovanni@research.telcordia.com
[3] UCLA
rafail@cs.ucla.edu
[4] Università di Salerno
giuper@dia.unisa.it

Abstract. We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword "urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word "urgent" is a keyword in the email without learning anything else about the email. We refer to this mechanism as *Public Key Encryption with keyword Search*. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We define the concept of public key encryption with keyword search and give several constructions.

1 Introduction

Suppose user Alice wishes to read her email on a number of devices: laptop, desktop, pager, etc. Alice's mail gateway is supposed to route email to the appropriate device based on the keywords in the email. For example, when Bob sends email with the keyword "urgent" the mail is routed to Alice's pager. When Bob sends email with the keyword "lunch" the mail is routed to Alice's desktop for reading later. One expects each email to contain a small number of keywords. For example, all words on the subject line as well as the sender's email address

* Supported by NSF and the Packard foundation.
** Partially supported by a gift from Teradata. Preliminary work done while visiting Stanford and while at Telcordia.
*** Part of this work done while visiting DIMACS. Work supported by NoE ECRYPT.

C. Cachin and J. Camenisch (Eds.): EUROCRYPT 2004, LNCS 3027, pp. 506–522, 2004.

$PK_1^{(c)}$ for some random $1 \leq c \leq q$. Next, $\mathcal{B}$ picks random $M_1, \ldots, M_{c-1}$, $M_{c+1}, \ldots, M_q \in \{0,1\}^s$ and sets $M_c = M$. Let $M' = M_1 \oplus \cdots \oplus M_q$. Algorithm $\mathcal{B}$ defines the following hybrid tuple:

$$R = \Big(M',\ E[PK_0^{(1)}, M_1],\ \ldots,\ E[PK_0^{(c-1)}, M_{c-1}],\ C,\ E[PK_1^{(c+1)}, M_{c+1}],\ \ldots,\ E[PK_1^{(q)}, M_q] \Big)$$

It gives R as the challenge PEKS to algorithm $\mathcal{A}$. Algorithm $\mathcal{A}$ eventually responds with some $b' \in \{0,1\}$ indicating whether R is $\mathsf{PEKS}(A_{pub}, W_0')$ or $\mathsf{PEKS}(A_{pub}, W_1')$. Algorithm $\mathcal{B}$ outputs b' as its guess for b. One can show using a standard hybrid argument that if $\mathcal{B}$ does not abort then $|\Pr[b = b'] - \frac{1}{2}| > \epsilon/q^2$. The probability that $\mathcal{B}$ does not abort at a result of a trapdoor query is at least $1 - (tq/d)$. The probability that $\mathcal{B}$ does not abort as a result of the choice of words W_0', W_1' is at least $(q/d)^2$. Hence, $\mathcal{B}$ does not abort with probability at least $1/poly(t, q, d)$. Repeatedly running $\mathcal{B}$ until it does not abort shows that we can get advantage ϵ/q^2 in breaking the source indistinguishability of (G, E, D) in expected polynomial time in the running time of $\mathcal{A}$. □

4 Construction Using Jacobi Symbols

Given the relation between Identity Based Encryption and PEKS it is tempting to construct a PEKS from an IBE system due to Cocks [3]. The security of Cocks' IBE system is based on the difficulty of distinguishing quadratic residues from non-residues modulo $N = pq$ where $p = q = 3 \pmod 4$.

Unfortunately, Galbraith [11] shows that the Cocks system as described in [3] is not public-key private in the sense of Bellare et al. [1]. Therefore it appears that the Cocks system cannot be directly used to construct a PEKS. It provides a good example that constructing a PEKS is a harder problem than constructing an IBE.

5 Conclusions

We defined the concept of a public key encryption with keyword search (PEKS) and gave two constructions. Constructing a PEKS is related to Identity Based Encryption (IBE), though PEKS seems to be harder to construct. We showed that PEKS implies Identity Based Encryption, but the converse is currently an open problem. Our constructions for PEKS are based on recent IBE constructions. We are able to prove security by exploiting extra properties of these schemes.

Acknowledgments

We thank Glenn Durfee for suggesting the use of H_2 in the construction of Section 3.1. We thank Yevgeniy Dodis, David Molnar, and Steven Galbraith for helpful comments on this work.

A Comparative Survey on Symmetric Key Encryption Techniques

Monika Agrawal
Department Of Computer Science
Shri ShankaraCharya Institute Of Technology & Management
Bhilai, India
monika.agrawal1986@gmail.com

Pradeep Mishra
Department Of Computer Science
Shri ShankaraCharya College Of Engineering & Technology
Bhilai, India
pradeepmishra4u@gmail.com

***Abstract*— Nowadays, the use of internet are growing increasingly across the world, security becomes a prime concern of issue for the society. Earlier security was a major issue for military applications but now the area of applications has been enhanced since most of the communication takes place over the web. Cryptography is an area of computer science which is developed to provide security for the senders and receivers to transmit and receive confidential data through an insecure channel by a means of process called Encryption/ Decryption. Cryptography ensures that the message should be sent without any alterations and only the authorized person can be able to open and read the message. A number of cryptographic techniques are developed for achieving secure communication. There are basically two techniques of cryptography- Symmetric and Asymmetric. This paper presents a detailed study of most of the symmetric encryption techniques with their advantages and limitations over each other.**

Keywords- Symmetric Encryption; Asymmetric Encryption; Cipher Text; Plain Text; Key

I. Introduction

In today's corporate world where access to information in lesser time is required with the goal of running the enterprise smoothly and efficiently, it is very important to give right information to right people at right time. What actually the information has been sent should be the same information been received. Suppose one person is sending an important file to the other person who is sitting at some other site office then the message passes through an insecure channel and may be possible that anyone in the middle can retrieve the message and modify it and then passes it to the destination. This will lead to many undesirable side-effects and the company may suffer a big loss in economical terms. Cryptography plays a very vital role in keeping the message safe as the data is in transit. It ensures that the message being sent at one end remains confidential and should be received only by the intended receiver at the other end.

Cryptography converts the original message in to non readable format and sends the message over an insecure channel. The people who are unauthorized to read the message try to break the non readable message but it is hard to do it so. The authorized person has the capability to convert the non readable message to readable one.

A. Basic Terms Used in Cryptography

- Plain Text
  The original message that the person wishes to communicate with the other is defined as Plain Text. In cryptography the actual message that has to be send to the other end is given a special name as Plain Text. For example, Alice is a person wishes to send "Hello Friend how are you" message to the person Bob. Here "Hello Friend how are you" is a plain text message.
- Cipher Text
  The message that cannot be understood by anyone or meaningless message is what we call as Cipher Text. In Cryptography the original message is transformed into non readable message before the transmission of actual message. For example, "Ajd672#@91ukl8*^5%" is a Cipher Text produced.
-

There are total 16 rounds of data encryption [8]. In each round a 32 bit subkey is XORed with leftmost 32 bits of plaintext and the result is then passed to the F function of Blowfish. This result becomes rightmost 32 bits for the next round and the output of F function is XORed with the original rightmost 32 bits of plaintext becomes leftmost 32 bits for the next round and so on.

1. Divide X into two 32-bit halves: XL, XR
2. For i = 1 to 16
   XL = XL ⊕ Pi
   XR=F(XL) ⊕ XR
   Swap XL and XR
3. Swap XL and XR (Undo the last swap)
4. XR=XR ⊕ P17
5. XL = XL ⊕ P18
6. Concatenate XL and XR

Figure 2. Blowfish Encryption Algorithm

The F function is the kernel and distinguishing feature of Blowfish and is applied as follows [10].First Divide XL (32 Bits) into four 8-bit quarters: a, b, c, and d. Then apply the formula

$$F(XL)=\{(S1[a] + S2[b]) \oplus S3[c]\} + S4[d])\} \qquad (3)$$

where + means addition modulo 2^32 , and⊕ means exclusive OR and S1, S2, S3, S4 are four substitution boxes.

The key of the Blowfish algorithm is 448 bits, so it requires 2^448 combinations to examine all keys [11]. The advantage of blowfish algorithm is that it is simple to implement since all operations carried out are XOR and addition. Moreover the speed of encryption and decryption are also known to be faster than other popular existing algorithms [9].

IV. Comparison

A comparison of popular encryption algorithms based on block size, key size, number of rounds and attacks if occurred is shown on Table II.

TABLE II. Comparison of DES, Triple DES, AES and Blowfish algorithm

	Symmetric Encryption Algorithms			
	DES	***TDES***	***AES***	***BLOWFISH***
Block Size	64 bit	64 bit	128 bit	64 bit
Key size	56 bit	168 bit	128,192, 256 bit	32-448 bit
Created By	IBM in 1975	IBM in 1978	Joan Daeman in 1998	Bruce Schneier in 1998
Algorithm Structure	Fiestel Network	Fiestel Network	Substitution Permutation Network	Fiestel Network
Rounds	16	48	9,11,13	16
Attacks	Brute Force Attack	Theoretically possible	Side Channel Attacks	Not Yet

The security of any algorithm is highly based on the length of key being used. In the above table it is clear that the key size of blowfish algorithm is high and that of DES is lesser. Hence it can be said that security of Blowfish is far better than the other algorithms. Also DES and other algorithms are vulnerable to possible attacks but Blowfish algorithm has not been cracked till date.

V. Conclusion

This paper gives a detailed study of the popular symmetric key encryption algorithms such as DES, TRIPLE DES, AES, and Blowfish. Symmetric Key algorithms run faster than Asymmetric Key algorithms such as RSA etc and the memory requirement of Symmetric algorithms is lesser than Asymmetric encryption algorithms. Further, the security aspect of Symmetric key encryption is superior than Asymmetric key encryption. The comparison table of popular encryption algorithms clearly shows the supremacy of Blowfish algorithm over DES, AES and Triple DES on the basis of key size and security. The F function of Blowfish algorithm provides

Double Layer Encryption Algorithm Key Cryptography for Secure Data Sharing in Cloud

Dr.D.Usha, M.Subbbulakshmi

Abstract— Cloud technology is extremely beneficial and valuable in present new technological epoch, where a person uses the remote servers and the internet to provide and preserve data as well as applications. Such application in revolve can be used by the users via the cloud infrastructure without any installation. Moreover, the users' data files can be accessed and manipulated from any other computer using the internet services. Despite the flexibility of data and application accessing and usage that cloud computing environments provide, there are many questions still coming up on how to achieve a trusted environment that guard application and data in cloud from unauthorized intruders. Since this paper propose a new era of key cryptography double layer encryption to make the cloud model more secure and trust worthy and a lot of work is being done regarding this. This paper aims to suggest an approach which is a double layer encryption method to ensure security in cloud. It is based on a popular cryptography algorithm RSA which is a relatively novel technique. This scheme resolves key escrow difficulty and data expose problem by RSA algorithm of public key cryptography approach. In this proposed double layers encryption schemes, the data will be extremely secured while protecting and sharing in cloud environment. This scheme not only makes full use of the great processing skill of cloud computing but also can efficiently ensure cloud data privacy and security.

Index Terms— Cloud Storage, Security, Privacy, RSA, Encryption, Cryptography, Double layer encryption.

1 Introduction

Nowadays, the technology has been developed based on the human requirements in the world. Lots of technologies are invented today and each one serves to people in different ways. This technology requires the resources like hardware, software for the effective utilization. From the efficient use it is processed with massive amount of data. The amount of data to handle in this world is completely panic. This situation brings us into a solution cloud computing. Cloud computing is a model for enabling convenient ubiquitous and on demand network access to a common pool of configurable computing resources. Storage of data becomes a main concern in the technical world. The amounts of data are easier to store and maintain with the help of cloud data storage services. It helps to store any large size of the data at different storage positions. Each position is operated in independent way. Business users are also being attracted by cloud storage due to its too many benefits, including lower cost, greater agility, and better resource utilization. However, while enjoying the quality of being useful, easy of distribution data via cloud storage, users are also increasingly worried about inadvertent data leaks in the cloud. Such data seep out caused a malicious. A misbehaving cloud operator, can usually lead to behave badly break or fail to observe of personal privacy or business secrets (e.g., the recent high probe incident of a famous person photos being leaked in iCloud). To address users relate to protect potential data leaks in cloud storage, a prevalent way of dealing with a circumstances is for the data owner to encrypt all the data before upload to cloud. In common privacy refers the condition or state of hiding the presence or view. There is a need to attain this state in the places where the private things are used such as data and files. In cloud data storage the privacy is needed to attain for the data, user identity and on controls. Violation of privacy leads to major failure in the system. To maintain the data privacy, it is possible for a successful deployment and usage of any service. Cloud computing is actually a combination of various traditional computing techniques like grid computing ,distributed computing, virtualization , load balancing ,etc. It combines the functionalities of all these and is evolve as a new model on which everyone can rely for everything. Cryptography is an antique art of hiding data to protect it from malicious users. The information to be sent over the network called plaintext is converted into unreadable form called cipher text by some encryption algorithm. On the receiver's side, the original data can be recovered from the cipher text by applying a decryption algorithm on it. Several algorithms are used to encrypt and decrypt the secret information. These are broadly classified as symmetric and asymmetric methods. Symmetric methods use the similar key for encryption and decryption while asymmetric methods use two keys (private key and public key).

In this paper the main task is to maintain the security and privacy for the data that is uploaded and the challenging problem is to allow only the authorized users to access the data from the cloud. Then the unauthorized users should not be permitted to access the data. In this paper multi-encryption system is proposed to provide the extra security for the data. The double-layer-encryption is performed by the data owner. If the user has to access the data then double-layer decryption should be done so that the safety and privacy of the data is high. The challenging problem is to monitor the attackers. Privacy of the users is taken in to account and only the authorized users are permitted to access or download the data to the cloud. Cloud Admin issues the token and maintains the keys of the users.

2 Literature Survey

1. An Efficient Certificate-less Encryption for Secure Data Sharing in Public Clouds

Decrypt and Re-Decrypt: The user decrypt and re-decrypt encrypted data using the secret key.

4.1 RSA Algorithm

The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977. The RSA cryptography is the mainly-used public key cryptography algorithm in the world. It can be used to encrypt a message using public key and decrypt a message using secret key without the need to exchange key. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the complexity of factoring huge integers.

Algorithm Steps

The RSA algorithm consists of three steps: key generation, encryption and decryption.

Key Generation

A key is a part of information that determines the efficient output of a cryptographic algorithm. Lacking of key, the algorithm would be worthless. In encryption, a key specifies the particular conversion of plaintext into ciphertext, or vice versa during decryption. There are two keys in RSA, i.e. Public key and Private Key. The public key is known to everyone and is used for encrypting the messages; these messages can be decrypted only using the private key.

- Choose two distinct prime numbers m and n.
- Find t such that t = p q. t will be used as the modulus for both the public and private keys.
- Find the totient of t, $\phi(t) = (m-1)(n-1)$.
- Choose an e such that $1 < e < \phi(t)$, and such that e and $\phi(t)$ share no divisors other than 1 (e and $\phi(t)$ are relatively prime). e is kept as the public key exponent.
- Determine d (using modular arithmetic) which satisfies the congruence relation
  $de \equiv 1 \pmod{\phi(t)}$.
- The public key is the pair (en, t).
- The private key is the pair (de, t).

Encryption

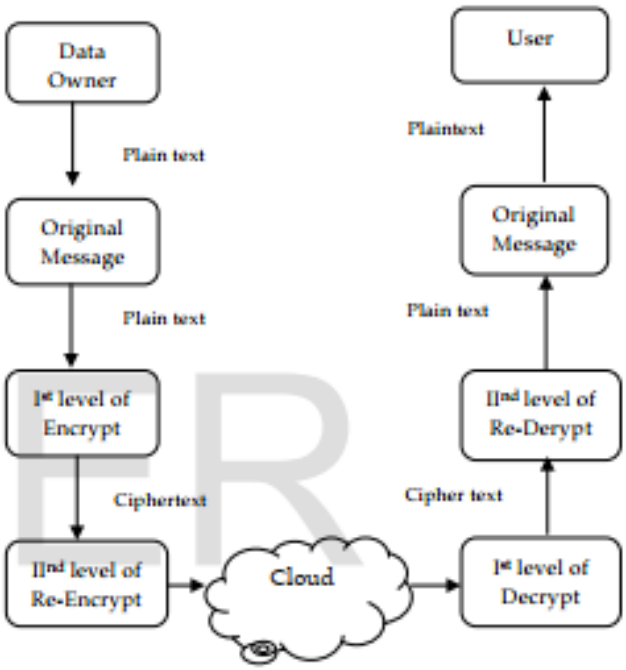
- Data owner transmits his/her public key (modulus n and exponent e) to user, keeping his/her private key secret.
- The data owner, first converts Msg to an integer such that $0 < x < y$ by using agreed upon reversible protocol known as a padding scheme.
- Data owner public key information, the ciphertext ct corresponding to $ct = xe \pmod{y}$.
- Data owner upload message "Msg" in ciphertext, or c, to cloud storage.

Decryption

- User recovers x from ct by using his/her private key exponent, d, by the computation $x = ctd \pmod{y}$.

Given m, user can recover the original message "Msg" by reversing the padding scheme.

4.2 System Model of Double Layer Encryption



4.3 Explanation of System Model

- **Plain Text** : Plaintext means the readable format text (Original Message)
- **Key (Public key)** : The original message is encrypted using public key
- **Encryption Process:** Encryption means the process of converting plaintext into ciphertext. Encryption process contains two level of encryption. In Ist level encryption, the plaintext is converted into ciphertext and IInd level encryption, the ciphertext is converted into ciphertext.
- **Encrypted Ciphertext:** The result of encryption phase (unreadable format).
- **Key (Private Key):** The ciphertext is decrypted using private key.
- **Decryption process:** Decryption means the process of converting ciphertext into plaintext. Decryption process contains two level of decryption. In Ist level decryption, the ciphertext is converted into ciphertext

On Ends-to-Ends Encryption

Asynchronous Group Messaging with Strong Security Guarantees

July 1st, 2017

Katriel Cohn-Gordon
University of Oxford

Cas Cremers
University of Oxford

Luke Garratt
University of Oxford

Jon Millican
Facebook

Kevin Milner
University of Oxford

ABSTRACT

In the past few years secure messaging has become mainstream, with over a billion active users of end-to-end encryption protocols through apps such as WhatsApp, Signal, Facebook Messenger, Google Allo, Wire and many more. While these users' two-party communications now enjoy very strong security guarantees, it turns out that many of these apps provide, without notifying the users, a weaker property for *group* messaging: an adversary who compromises a single group member can intercept communications indefinitely.

One reason for this discrepancy in security guarantees is that most existing group messaging protocols are fundamentally *synchronous*, and thus cannot be used in the asynchronous world of mobile communications. In this paper we show that this is not necessary, presenting a design for a tree-based group key exchange protocol in which no two parties ever need to be online at the same time. Our design achieves strong security guarantees, in particular including post-compromise security.

We give a computational security proof for our core design as well as a proof-of-concept implementation, showing that it scales efficiently even to large groups. Our results show that strong security guarantees for group messaging are achievable even in the modern, asynchronous setting, without resorting to using inefficient point-to-point communications for large groups. By building on standard and well-studied constructions, our hope is that many existing solutions can be applied while still respecting the practical constraints of mobile devices.

KEYWORDS

end-to-end encryption, group messaging, tree Diffie-Hellman, security protocols, computational proof, verification

1 INTRODUCTION

The level of security offered by secure messaging systems has improved substantially over recent years; for example, WhatsApp now provides end-to-end encryption for its billion active users, based on Open Whisper Systems' Signal Protocol [28, 34], and the Guardian publishes Signal contact details for its investigative journalism teams [17]. An important constraint of modern messaging systems, compared to related protocols such as those used for key exchange, is that they must allow for *asynchronous communication*: Alice must be able to send a message to Bob even if Bob is currently offline. Typically, the encrypted message is temporarily stored on a (possibly untrusted) server, to be delivered to Bob once he comes online again.

This asynchronicity constraint implies that standard solutions to achieve, e.g., perfect forward secrecy, such as a Diffie–Hellman (DH) key exchange, do not apply directly. This has driven the development of novel techniques to achieve perfect forward secrecy without interaction, for example using sets of "prekeys" [27] that Bob uploads to a server, essentially serving as precomputed DH keys, or by using puncturable encryption [16].

Moreover, some modern messaging protocols offer a property called post-compromise security (PCS) [9], often referred to as "future secrecy" or "self-healing": even after Alice's device is entirely compromised by an adversary, revealing her long-term key and potentially all random values generated so far, she may be able to later regain secure communications with others, as long as she has one exchange with them in which the adversary does not interfere. PCS limits the scope of a compromise, forcing an adversary to act as a permanent man-in-the-middle if they wish to exploit knowledge of a long-term key. Thus far, PCS-style properties have only been proven for point-to-point protocols.

In practice however, point-to-point communication does not suffice for real-world messaging applications, in which group and multi-device messaging are often important features. In theory, it is easy to solve this: Alice uses the point-to-point protocol with each of her communication partners. However, as group sizes become larger, this leads to inefficient systems in which the bandwidth and computational cost for sending a message grows linearly with the group size (as each recipient gets their own, differently encrypted, copy of the message). In many real-world scenarios, this inefficiency can be problematic, especially in areas with restricted bandwidth or high data costs (e.g., 2G networks in the developing world). The 2015 State of Connectivity report by internet.org [18] lists affordability of mobile data as one of the four major barriers to global connectivity, with a developing-world average monthly data use of just 255 MB/device.

Instead of using a point-to-point protocol with each group member, a theoretical alternative is to use a group protocol [6, 7, 11, 20–22, 25]. These typically use tree structures based on DH keys to combine the participants' individual keys into a group key. This reduces both the computational effort and bandwidth required to send a message, as the sender sends only one copy of each message encrypted under the group key. However, such protocols are in general not asynchronous, and do not consider post-compromise security—they do not make any guarantees after the adversary completely compromises a participant.

K.C-G. thanks the Oxford CDT in Cyber Security for their support.

Table 3: Times in milliseconds for our implementation to perform various tree operations. All computation was performed on a 2016 Apple MacBook Pro, and results are the average of 5 benchmark runs. In *construct tree*, the initiator fetches public keys for each group member, and follows the unauthenticated algorithm in Section 5 to build a complete DH tree. In *import tree*, responders use their private key to compute the first stage key. In encrypt (initiator), the initiator derives new chain keys from the first stage key, and uses them to encrypt messages, decrypted by the responder in *decrypt (responder)*. In *encrypt (responder)*, the responder performs a tree update as per Section 5 to derive a new stage key and then message key, encrypting a message decrypted in *decrypt (initiator)*.

group size	construct tree (initiator)	import tree (responder)	encrypt (initiator)	decrypt (responder)	encrypt (responder)	decrypt (initiator)
2	1.8	0.4	0	0.8	0	0.4
7	5.8	0	0.2	0.8	0.2	0
127	90.6	0	0.2	1.6	0	0
32,768	25,187.8	20.6	0.6	4.2	0.4	0
100,000	77,022.8	248.8	0.6	3.8	0.4	0

8 EXTENSIONS

We here remark on various possible extensions to our designs. In general, because we use standard tree-DH techniques, much of the existing literature is directly applicable. This means that we can directly apply well-studied techniques which do not require interactive communication rounds.

Sender-specific authentication. As early as 1999, Wallner et al. [33] pointed out the issue of "sender-specific authentication": in a system which derives a shared group key known to all members, there is no cryptographic proof of *which* group member sent a particular message. Various works have discussed such proofs; the most common design is to assign to each group member a signature key with which they sign all their messages. We remark that it is easy to extend our design with such a system.

In particular, if an agent generates and broadcasts new signature keys together with their new leaf keys, signing the new key with the old one, then we conjecture that they will achieve a form of authentication even post-compromise.

Dynamic groups. We refer the reader to e.g. [21] for a summary of previous work on dynamic groups. In general, since we build on tree-based ideas, our design can support join and leave operations using standard techniques.

We remark in particular that these operations can be done *asynchronously* using a design similar to the setup keys in Section 5.1. Specifically, Alice can add Ted as a sibling to her own node in the tree by performing an operation similar to the initial tree setup, generating an ephemeral key and performing a key update which replaces Alice's leaf with an intermediate node whose children are Alice and Ted. With the cooperation of other users in the tree, Alice can add Ted *anywhere*, allowing her to keep the tree balanced.

Multiple Devices. One important motivation for supporting group messaging is to enable users to communicate using more than one of their own devices. By treating each device as a separate group member, our design of course supports this use case. However, the tree structure can be optimised for this particular scenario: all of Alice's devices can be stored in a single subtree, so that the "leaves" of the group tree are themselves roots of device-specific trees. This has two particular benefits.

First, in her own time Alice can execute the group key agreement protocol just between her devices, and use the resulting shared secret as an ephemeral prekey or "pretree". This allows other group members to retrieve a single key for Alice, instead of adding all of her devices as separate entities in the group tree. If most users have multiple devices, this can significantly reduce the size of group trees.

Second, when any of Alice's devices performs a key update, the other group members only need to know the public keys from the root of Alice's subtree to the root of the group tree. In particular, Alice does not need to broadcast to the group the set of her devices or the metadata about which device is performing a key update. Thus, she can maintain end-to-end encryption between all of her devices while still keeping the list private.

Chain keys. The Signal protocol introduced the concept of *chain keys* to support out-of-order message receipt as well as a fine-grained form of forward secrecy. Instead of using a shared secret to encrypt messages directly, Signal derives a new encryption key for each message by applying a key derivation function to the current key, generating a new chain key in the process.

The shared secret derived by our group key exchange can be directly used as the start of a key chain. Indeed, our implementation derives its message keys from a hash chain, ensuring that each key is only ever used once as well as providing a form of forward secrecy after compromise of a chain key.

9 CONCLUSION

While modern messaging applications can offer strong security guarantees, they typically only do this for two-party communications. If another person is added to the group, the effective security guarantees are decreased, without notifying the users of this security degradation.

In this paper, we combined techniques from synchronous group messaging with modern guarantees from asynchronous messaging. Our resulting asynchronous design combines the bandwidth benefits of group messaging with the strong security guarantees of modern point-to-point protocols. This paves the way for modern messaging applications to offer the same type of security for groups that they are currently only offering for two-party communications.

Scientific Research Publishing

Journal of Information Security, 2018, 9, 95-99
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger

Robert E. Endeley

Capitol Technology University, Laurel, MD, USA
Email: reendeley@captechu.edu

How to cite this paper: Endeley, R.E. (2018) End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger. *Journal of Information Security*, **9**, 95-99.
https://doi.org/10.4236/jis.2018.91008

Received: December 22, 2017
Accepted: January 20, 2018
Published: January 23, 2018

Abstract

The ubiquity of instant messaging services on mobile devices and their use of end-to-end encryption in safeguarding the privacy of their users have become a concern for some governments. WhatsApp messaging service has emerged as the most popular messaging app on mobile devices today. It uses end-to-end encryption which makes government and secret services efforts to combat organized crime, terrorists, and child pornographers technically impossible. Governments would like a "backdoor" into such apps, to use in accessing messages and have emphasized that they will only use the "backdoor" if there is a credible threat to national security. Users of WhatsApp have however, argued against a "backdoor"; they claim a "backdoor" would not only be an infringement of their privacy, but that hackers could also take advantage of it. In light of this security and privacy conflict between the end users of WhatsApp and government's need to access messages in order to thwart potential terror attacks, this paper presents the advantages of maintaining E2EE in WhatsApp and why governments should not be allowed a "backdoor" to access users' messages. This research presents the benefits encryption has on consumer security and privacy, and also on the challenges it poses to public safety and national security.

Keywords

Instant Messaging, WhatsApp, End-to-End Encryption, National Security, Privacy

1. Introduction

The world is ever changing due to the advancement in the realm of science and technology, and these days it seems hard to escape the presence of technology in

question." [9]. WhatsApp Inc. has since responded to this claim, saying that the feature in question is a design tradeoff, meant to prevent users from losing their messages if they switch phones or reinstall the app.

3. Conclusion

While a majority of countries would favor some kind of restriction on access to unrecoverable encryption, there is no global consensus, and the likely outcome is a hodgepodge of national policies. According to [10], "Our research suggests that the risk to public safety created by encryption has not reached the level that justifies restrictions or design mandates". Lewis *et al.* further went on to say, "The encryption issue that law enforcement faces, while frustrating, is currently manageable". Communications privacy is a key element of human rights in the digital era, and developments affecting it ought to be reported. Ultimately, removing WhatsApp E2EE would not be the solution, as criminals could create their own, similar software that would allow them to communicate securely, while ordinary users would lose the ability to send genuinely private messages [6]. Maintaining E2EE in WhatsApp without an encryption backdoor guarantees genuine privacy in conversations between individuals and group chats. Voice conversations through WhatsApp messenger feel more natural; users are assured that no one is eavesdropping on their conversations, and conversations thus tend to feels more like a face-to-face conversation.

References

[1] Yeboah, J. and Ewur, G. (2014) The Impact of WhatsApp Messenger Usage on Students Performance in Tertiary Institutions in Ghana. *Journal of Education and Practice*, **5**, 157-164.

[2] Sarker, G.R. (2015) Impact of WhatsApp Messenger on the University Level Students: A Sociological Study. *International Journal of Natural and Social Sciences*, **2**, 118-125.

[3] Jisha, K. and Jebakumar (2014) A Trend Setter in Mobile Communication among Chennai Youth. *IOSR Journal of Humanities and Social Science* (*IOSR-JHSS*), **19**, 01-06.

[4] Central Intelligence Agency (2017) The World Factbook. Country Comparison, Internet Users. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html

[5] Pascual, A. (2013) Data Breaches Becoming a Treasure Trove for Fraudsters, 2013 Identity Fraud Report. https://www.javelinstrategy.com/coverage-area/2013-identity-fraud-report-data-breaches-becoming-treasure-trove-fraudsters

[6] Michalas, A. (2017) How WhatsApp Encryption Works—And Why There Shouldn't Be a Backdoor. The Conversation. https://theconversation.com/how-whatsapp-encryption-works-and-why-there-shouldnt-be-a-backdoor-75266

[7] District Attorney New York County (2005) Going Dar: Encryption, Technology and the Balance between Public Safety and Privacy. District Attorney New York County,

3rd International Conference on Computing: Communication, Networks and Security (IC3NS-2018)
Volume: 4 Issue: 3
ISSN: 2454-4248
39 – 42

Whatsapp End-To-End Encryption

Aashi Jain, Aastha Gupta, Sonal Soni

College of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh (Rajasthan), 332311, India.

jainaashi1995ja@gmail.com, aasthag1811@gmail.com, soni7sonal@gmail.com

+91 9457197007, +91 8168377718, +91 7726802076

Abstract: WhatsApp is a platform widely used for online communication and for exchanging text as well as audio and video messages. It is of great importance that these messages be secure. To maintain the privacy of the users WhatsApp uses end-to-end encryption technique and uses various protocols and functions like the Curve25519 function and STA-256 algorithm. WhatsApp uses the Signal Protocol which was developed by the Open Whisper System. Using end-to-end encryption, any middle man or even WhatsApp server cannot retrieve the messages and thus the privacy and security of the users is maintained.

Keywords: *Identity Key, Signed Pre Key, One Time Pre Key, Signal Protocol, Curve 25519 function, ECDH key agreement protocol*

*****

I. INTRODUCTION

WhatsApp is a widely used social media tool to interact with people; it is widely used to chat, message, voice call or video call with our near and dear ones. It is also used to share important and private data among people. It is of great importance to ensure that all this data is secure and is available only to the sender and the receiver and no one else can access this data. To ensure this, WhatsApp uses end-to-end encryption technique to ensure that nobody can hack the data and even WhatsApp itself cannot access the data.

II. END-TO-END ENCRYPTION

End-to-end encryption [1] means that the message or data sent by a person to another person can only be understood by the two of them. No third person can understand that data even if he gets access to the same. The message (be it audio, video or text) travels in an encrypted form and only the recipient is able to decrypt it. Even the Internet Service Provider cannot get access to the message. It is of importance to ensure the security and the privacy of the end users. If any communication app is encrypted, then it does not mean that the owner company cannot view the messages. The company itself has the key to decrypt the messages and therefore it does not completely keep the privacy of the users intact. End-to-end encryption on the other hand ensures that the users are completely secured and even the owner company cannot view the messages due to lack of the keys required. The role of the company servers is to simply forward the encrypted message to the receiver.

III. METHODS

WhatsApp uses the Signal Protocol [4] (earlier known as TextSecure Protocol) to implement end-to-end encryption. This protocol works on the concept of keys. Each user has his own private key that is made at time of installation of the app on the user's phone. The corresponding recipient has the user's public key through which he can decrypt the messages encrypted by the sender. Even the WhatsApp server does not have the private key of the user so it cannot peer in the conversation of the two users. Moreover, to enhance the security, each message is encrypted by its own key and cannot be hacked even if the hacker finds the key as it changes time to time and is different for each message.

The keys used in the encryption and decryption process are-

1. Identity Key- This is the user's private key generated at the time of app installment and it remains the same. This key is not available to any other user or even the WhatsApp server.
2. Signed Pre Key- This key is also generated at the install time and the identity key is used to sign this key.
3. One Time Pre Key- This key is generated for one time use and then it is deleted from the server's memory.

The above three keys are all public and are used to encrypt the message which can only be decrypted by the receiver who has the private key. As the private key is present only at the device of the user that is why no other device/user can decrypt the message and it is completely secure.

These keys are generated with the help of the Curve25519 function of the ECDH (Elliptic Curve Diffie Helman) key agreement protocol. The One Time Pre Key is deleted once a recipient sends a message to the user.

Now, for each message, a message_secret is generated. This

ISSN: 2454-4248
39 – 42

can listen to the talks or video chats between two users. To initiate a voice or video call, first of all there has to be a session. If the session does not already exist between the initiator and the recipient, then it has to be established by following the same procedure. Then, to call the recipient, the initiator has to generate a SRTP [5] master secret. This master secret is sent to the recipient along with an encrypted message which tells the recipient about the incoming call. If the recipient accepts the call, then the SRTP encrypted call ensues.

VIII. STATUS ENCRYPTION

In addition to the communication between two or more users, WhatsApp also encrypts the status and the profile picture of the clients. This is done in a similar manner as the group messages encryption. Once the status is updated by a client, then all the other clients who are authorized to view the status receive the keys and they can decrypt the status as well as the profile picture with the help of their respective private keys and the public keys of the client whose status has to be viewed.

VULNERABILITY

Suppose one user sends a message to other user who is currently offline. WhatsApp will temporarily store the message in the unencrypted form to send it to the receiver when he comes online. In the meantime, it is possible that some third person convinces the WhatsApp server that he is the receiver by either accessing the device or by hacking the GSM network [7] so that he can get the OTP (One Time Password) that WhatsApp would use to make sure that the message belongs to the receiver. Also, the government can persuade WhatsApp to get access to the messages of the clients and therefore behave as some other user to get access to his private messages and data.

IX. PROTOCOLS USED

Signal Protocol- This protocol was developed by the Open Whisper System. It is used to provide end-to-end encryption to text, audio or video messages. It uses the Curve 25519, AES-256 and HMAC-SHA256 algorithms.

ECDH Protocol- It is a key agreement protocol. It defines how keys should be generated and exchanged between end users or two or more parties. It takes the input as two keys and gives the output in the form of a secret key.

SRTP Protocol- Secure Real-Time Transport Protocol is mainly used for enhanced security features. It is widely used to secure the VOIP (Voice Over Internet Protocol) communications.

X. DISADVANTAGES

Along with all the advantages of end-to-end encryption, there are some disadvantages also. Due to end-to-end encryption, there is no watch on the data that is being circulated. It has become difficult to keep a watch on the activities of the anti-social people who misuse the social media to spread hatred messages.

XI. CONCLUSION

The main techniques that WhatsApp uses for end-to-end encryption are-

1. ECDH- This protocol is used to establish a shared secret by two parties over an unsecure channel
2. HKDF- It is the key derivation function. It derives Root key and Chain key from the master_secret.
3. HMAC-SHA256 (Hash-based Message Authentication Code) - It gives the output as a hash value from the input provided.
4. AES256 (Advanced Encryption Standard) - It is a symmetric encryption algorithm. It gives the result as a downloadable text file

REFERENCES

[1]. end-to-end encryption (E2EE) (2015) Retrieved February 7, 2018 from http://searchsecurity.techtarget.com/definition/end-to-end-encryption-E2EE

[2]. Calvin Li, Daniel Sanchez, Sean Hua (2016) WhatsApp Security Paper Analysis Retrieved from

How WhatsApp encryption works – and why there shouldn't be a backdoor

28 March 2017, by Antonis Michalas

Credit: Shutterstock

A battle between national security and privacy is brewing. Governments and secret services are asking encrypted messaging services such as WhatsApp to allow them access to users' data. Most recently, in the wake of the March attack at Westminster, Amber Rudd, the UK home secretary, said it was unacceptable that the government couldn't read the encrypted messages of suspected terrorists.

The main argument behind this request is that access to messages will allow authorities to thwart future terror attacks. On the other hand, there are many ordinary people who use messaging apps for daily communication and this request would be a direct breach of their privacy. But this isn't the only problem – creating a way for the authorities to read encrypted messages would also make the system vulnerable to cyber attacks from criminals and other hackers, removing what makes it a secure way to communicate in the first place.

How does encryption work?

Encryption is simply a way for two or more users to exchange messages securely. Encryption algorithms are like a box with two locks. For example, if a user called Alice wants to send her friend Bob a secure message, she puts it in the box and locks it with her key. Then, she sends the locked box to her friend Bob, who can only open the box and read Alice's message if he has a valid key of his own.

But to be able to communicate with new users, you need a way of sharing keys that is still secure. To get over this, each user has what's called a public key that is available to anyone and proves the identity of the user, and a private key that stays with the user. Alice uses Bob's public key to lock the box, but it can only be unlocked with Bob's private key.

WhatsApp's system adds a further level of encryption, known as "perfect forward secrecy". This is like a second lock with a key that changes for every messaging session. When Alice wishes to send a message to Bob, she first generates a fresh session key, places it in the box and uses Bob's public key to lock it. She then sends it to Bob, who uses his private key to access the session key. The two of them can then start communicating securely using that session key known only to them to encrypt their messages.

This system guarantees that there is no single key that will give access to all the data sent between Alice and Bob in the past or future. In other words, even if a key is compromised, it will only unlock a few messages before it becomes useless.

However, WhatsApp's previous system meant that the company was able to access the keys and so in theory could easily unlock the messages, breaching Alice's privacy. Last year, the company introduced what's called "end-to-end encryption", which seems to have solved this problem. Alice and Bob now use keys that WhatsApp doesn't keep specific details of, meaning only Alice and Bob can unlock their messages.

What government wants

Breaking Message Integrity of an End-to-End Encryption Scheme of LINE*

Takanori Isobe[1] and Kazuhiko Minematsu[2]

[1] University of Hyogo, Japan. takanori.isobe@ai.u-hyogo.ac.jp
[2] NEC Corporation, Japan. k-minematsu@ah.jp.nec.com

Abstract. In this paper, we analyze the security of an end-to-end encryption scheme (E2EE) of LINE, a.k.a Letter Sealing. LINE is one of the most widely-deployed instant messaging applications, especially in East Asia. By a close inspection of their protocols, we give several attacks against the message integrity of Letter Sealing. Specifically, we propose forgery and impersonation attacks on the one-to-one message encryption and the group message encryption. All of our attacks are feasible with the help of an end-to-end adversary, who has access to the inside of the LINE server (e.g. service provider LINE themselves). We stress that the main purpose of E2EE is to provide a protection against the end-to-end adversary. In addition, we found some attacks that even do not need the help of E2E adversary, which shows a critical security flaw of the protocol. Our results reveal that the E2EE scheme of LINE do not sufficiently guarantee the integrity of messages compared to the state-of-the-art E2EE schemes such as Signal, which is used by WhatApp and Facebook Messenger. We also provide some countermeasures against our attacks. We have shared our findings with LINE corporation in advance. The LINE corporation has confirmed our attacks are valid as long as the E2E adversary is involved, and officially recognizes our results as a vulnerability of *encryption break*.

Key words: E2EE, LINE, key exchange, group message, authenticated encryption

1 Introduction

1.1 Background

An end-to-end encryption (E2EE) is a secure communication scheme for messaging applications where the only people who are communicating can send and read the messages, i.e. no other party, even service providers of communication system, cannot access to the cryptographic keys needed to encrypt the message, and decrypt the ciphertexts. After Snowden's revelation, the E2EE receives a lot of attentions as a technology to protect a user privacy from mass interception and surveillance of communications carried out by governmental organizations such as NSA (National Security Agency).

Apple first supported an E2EE scheme in their widely-deployed messaging application, iMessage, where a message that is compressed by gzip is encrypted by a sender's secret key and distributed with a digital signature for the guarantee of the integrity to the recipient. Unfortunately, several security flaws of the initial iMessage are pointed out in 2016 [26]. A Signal is a new E2EE protocol for instant messaging. The core of the Signal protocol has been adopted by WhatsApp, Facebook Messenger, and Google Allo. A novel technology called ratcheting key update structure enables advanced security properties such as perfect forward secrecy and so-called post-compromise security [21]. Since Signal is an open-source application and its source code for Android and iOS are available on Github [31], its security has been studied well from the cryptographic community [19, 20, 32].

LINE is one of the most widely-deployed messaging applications, especially in East Asia. The number of monthly active users of four key countries, namely Japan, Taiwan, Thailand and Indonesia is about 217 million in January 2017. Their market is still growing, and at the same time their applications are expanding such as banking, payment, shopping, music services. Indeed, it is currently a key platform for any IT services in these countries. For example, Japanese government recently launched a portal cite for management of Japanese social security number, called

* Full version of the paper published in the proceedings of ESORICS 2018

Due to the structural similarity, it may make sense to compare LINE-AE with a generic composition of CBC encryption using AES-256 and HMAC-SHA-256 in terms of security. Here, we assume a random 128-bit initial vector (IV) or salt, and HMAC output is truncated to 128 bits, and CBC and HMAC are composed in the encrypt-then-mac fashion, using independent keys. Given the composition is correctly done, following Krawczyk [28] and Namprempre et. al. [30], and the standard cryptographic assumptions on AES and (the compression function of) SHA-256, the composition CBC+HMAC has 64-bit security for privacy, which comes from the provable security of CBC encryption (where 64-bit security of CBC is from the collision probability among the inputs to AES), and 128-bit for authenticity from the security of HMAC [27]. The privacy bound of LINE-AE is 32 bits, and if the salt was 128 bits, it seems not hard to derive 64-bit privacy bound, which is largely equivalent to CBC+HMAC. On the contrary, it seems less trivial to derive the authenticity bound. We expect it is possible to derive one assuming the second preimage resistance of the 128-bit SHA′-256 hash function. Our attack of Section 6.2 supports this observation, since it essentially breaks the second preimage resistance of SHA′-256 using 2^d targets.

However, we stress that our attack against LINE-AE allows treading-off of offline and online computations, and needs only single forgery attempt to have a sufficiently high success probability. At the extreme case, we can attack LINE-AE using 2^{128} offline computation with single ciphertext and single forgery attempt, which seems not possible with CBC+HMAC using 256-bit keys.

7 Conclusion

In this paper, we have evaluated the security of the E2EE scheme of LINE, one of the popular messaging applications in East Asia, and proposed several practical attacks. We first showed impersonation and forgery attacks on the group messaging scheme by a malicious group member. Next, we presented the malicious key exchange attack on the one-to-one messaging scheme. Then, we evaluated the security of the authenticated encryption scheme used in the message encryption phase, and presented the forgery attack against the the authenticated encryption scheme by the E2E adversary. We discussed practicality and fesibility of our attacks by considering the use cases of LINE. As a result, we conclude that the E2EE scheme of LINE do not provide a sufficient level of security compared to the start-of-the-art E2EE schemes such as Signal, which is used by WhatApp and Facebook Messenger, and Apple's iMessage.

Acknowledgments The authors would like to thank the anonymous referees for their insightful comments and suggestions. We are also grateful to LINE corporation for the fruitful discussion and feedback about our findings.

References

1. FIPS PUB 197, Advanced Encryption Standard (AES), 2001. U.S.Department of Commerce/National Institute of Standards and Technology.
2. NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation, 2001. U.S.Department of Commerce/National Institute of Standards and Technology.
3. NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, 2007. U.S.Department of Commerce/National Institute of Standards and Technology.
4. NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, 2007. U.S.Department of Commerce/National Institute of Standards and Technology.
5. FIPS PUB 180-4, Secure Hash Standard, 2015. U.S.Department of Commerce/National Institute of Standards and Technology.
6. New generation of safe messaging: Letter Sealing. LINE Blog., 2015. `https://engineering.linecorp.com/en/blog/detail/65`.
7. LINE Enters Agreement with Japan's CAO for Mynaportal Interconnectivity, 2017. `https://linecorp.com/en/pr/news/en/2017/1771`.
8. Line Will Top 50 Million Users in Japan This Year. eMarketer, 2017. `https://www.emarketer.com/Article/Line-Will-Top-50-Million-Users-Japan-This-Year/1016207`.

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 4 April 2017, Page No. 20915-20919
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.20

A research Paper on Cryptography Encryption and Compression Techniques

Sarita Kumari
Research Scholar

Abstract

Data is any type of stored digital information. Security is about the protection of assets. Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, personal databases and websites. Cryptography is evergreen and developments. Cryptography protects users by providing functionality for the encryption of data and authentication of other users. Compression is the process of reducing the number of bits or bytes needed to represent a given set of data. It allows saving more data. Cryptography is a popular ways of sending vital information in a secret way. There are many cryptographic techniques available and among them AES is one of the most powerful techniques. The scenario of present day of information security system includes confidentiality, authenticity, integrity, non-repudiation. The security of communication is a crucial issue on World Wide Web. It is about confidentiality, integrity, authentication during access or editing of confidential internal documents.

Keywords : Data Encryption and decryption, Compression, Cryptography Concept, Security, Integrity.

Introduction

To secure the data, compression is used because it use less disk space (saves money), more data can be transfer via internet. It increase speed of data transfer from disk to memory. Security goals for data security are Confidential, Authentication, Integrity, and Non-repudiation. Data security delivers data protection across enterprise. Information security is a growing issue among IT organizations of all sizes. To tackle this growing concern, more and more IT firms are moving towards cryptography to protect their valuable information. In addition to above concerns over securing stored data, IT organizations are also facing challenges with everincreasing costs of storage required to make sure that there is enough storage capacity to meet the organization's current and future demands. Data compression is known for reducing storage and communication costs. It involves transforming data of a given format, called source message to data of a smaller sized format called code word. Data encryption is known for protecting information from eavesdropping. It transforms data of a given format, called plaintext, to another format, called cipher text, using an encryption key. Currently compression and encryption methods are done separately. Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. The growth of cryptographic technology has raised a number of legal issues in the information age. Cryptography's potential for use as a tool for espionage and sedition has led many governments to classify it as a weapon and to limit or even prohibit its use and export.

Data compression implies sending or storing a smaller number of bits. Compression is the reduction in size of data in order to save space or transmission time. Many methods are used for this purpose, in general these methods can be divided into two broad categories: Lossy and Lossless methods. Lossy Compression generally used for compress an images. In this original data is not identical to compressed data that means there is some loss e.g. Block Truncation Coding, Transform Coding, etc... Lossless Compression used for compress any textual data.

SUMMARY

Cryptography is used to ensure that the contents of a message are confidentiality transmitted and would not be altered. Confidentiality means nobody can understand the received message except the one that has the decipher key, and "data cannot be changed" means the original information would not be changed or modified.

REFERENCES

[1]Swarnalata Bollavarapu and Ruchita Sharma-" Data Security using Compression and Cryptography Techniques"

[2] Manoj Patil, Prof. Vinay Sahu—" A Survey of Compression and Encryption Techniques for SMS"

[3] Bobby Jasuja and Abhishek Pandya –" Crypto-Compression System: An Integrated Approach using Stream Cipher Cryptography and Entropy Encoding"

[4] https://msdn.microsoft.com/en-us/library/windows/desktop/aa381939(v=vs.85).aspx

[5] https://www.techopedia.com/definition/1773/decryption

[6]www.computerhope.com/jargon/d/decrypti.htm

[7] https://en.wikipedia.org/wiki/Cryptography

[8] https://www.techopedia.com/definition/25403/encryption-key

[9] http://searchsecurity.techtarget.com/definition/private-key

[10]https://www.tutorialspoint.com/cryptography/cryptography_tutorial.pdf

Social Authentication for End-to-End Encryption

Elham Vaziripour
Brigham Young University
elhamvaziripour@byu.edu

Mark O'Neill
Brigham Young University
mto@byu.edu

Justin Wu
Brigham Young University
justinwu@byu.edu

Scott Heidbrink
Brigham Young University
sheidbri@byu.edu

Kent Seamons
Brigham Young University
seamons@cs.byu.edu

Daniel Zappala
Brigham Young University
zappala@cs.byu.edu

1. INTRODUCTION

Over the last several decades, it has become increasingly important to secure data via end-to-end encryption. The Internet has evolved to provide security for connections, primarily using TLS (or SSL), but generally fails to provide true end-to-end encryption. While TLS and similar protocols encrypt data during transit, data at rest is often unprotected, residing in storage on a client or server machine in plaintext. Data in this state are susceptible to honest-but-curious service providers, hackers, physical theft, and coercive governments.

Generic public-key cryptography provides powerful mechanisms to enable end-to-end encryption, but providing good usability for these mechanisms is a challenging task for novice users—leading to the decades-long situation where "Johnny can't encrypt" [8, 7, 6]. The primary problems center on *user-to-user authentication* – authenticating users to each other by associating their identities with public keys. We have made significant progress authenticating web sites to users (via X509 certificates and associated authorities) and authenticating users to web sites (with passwords). Each of these have their challenges, but have at least been widely deployed. Authenticating users to one another, however, has seen relatively little adoption. Usable mechanisms for personal key management, key distribution, and key authentication are still largely open issues.

Significant progress has been made recently with the proliferation of secure messaging apps such as Signal and ChatSecure. These applications address the aforementioned issues in a variety of ways. First, operating primarily on mobile devices greatly mitigates key management problems, since users almost always have their mobile devices on their person. Second, these apps also store mappings between identities and keys and perform authentication on behalf of users, reducing the need for manual collection of key-identity pairs and authentication. During the first communication with another user, users are often advised to perform out-of-band validation of public keys (e.g. reciting public key fingerprints by voice through a phone call), but it is not clear how frequently this is done. Afterward, the verified keys are stored on their respective devices and future authentication proceeds automatically (and locally). Similar efforts have been made to provide usable, secure email. Efforts such as Private WebMail (PWM) [5] and Virtru handle generation and validation of keys on behalf of users by means of key escrow. There is some evidence that users prefer the convenience of automatic (but potentially less secure) methods rather than manual key exchange [1].

Despite the improvements these applications bring, some notable issues persist:

- **Key discovery:** Discovering the public key of a user may or may not be possible. For example, Signal uses phone numbers to look up identities for remote users, prohibiting contact with anyone whose phone number is unknown.
- **Key validation:** These apps rely on the user to verify the public key of their associates through some manual means. Such behavior is also not enforced, reducing the security of the system overall.
- **Generality:** These and similar applications are not general in two ways. First, these applications use specific mediums for communication rather than supporting key infrastructure for communication across arbitrary mediums (Signal uses SMS and ChatSecure uses XMPP). Second, the applications cannot be used from other devices such as laptops because their associated private keys rely on the mobile device only.
- **Trusted third parties:** Pwm and Virtru rely on a centralized server to verify that users own their respective email addresses and delivers private keys to users based on this third party authentication. While this assists greatly in application portability and key discovery, reliance on a trusted third party violates the true spirit of end-to-end encryption.

We need more work on user-to-user authentication solutions that are general and portable, automate key discovery, and bootstrap and automate key validation as much as possible. An identity and associated contacts should be portable across the devices and the applications a person uses. Key management should avoid reliance on trusted third parties, so that the security guarantees provided to users cannot be easily broken by governments or service providers.

This work was supported by the National Science Foundation under Grant No. CNS-1528022. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Symposium on Usable Privacy and Security (SOUPS) 2016, June 22–24, 2016, Denver, Colorado.

2. SOCIAL AUTHENTICATION

We propose that the issues of key discovery and validation can be solved by bootstrapping off the trust that already exists among users of online social networks (OSNs). Each OSN already provides users with long histories of posts, pictures, and personal communications from their contacts and provides authentication of its users (via password logon). By following a verified user on Twitter or accepting a friend request on Facebook, users are already making an authentication judgment. Thus if public keys were posted to and associated with users' various OSN accounts, an organic set of verified key-identity pairs could emerge. By querying keys for a user from multiple OSNs and checking for agreement, the application could enhance trust in a public key, as multiple OSNs would have be compromised or collude to present a believable false key. Such a system mitigates much of the manual key validation problem as users can rely on the robustness of multiple authorities vouching for the authenticity of a particular key. This mechanism also has an added benefit – users need only have some type of OSN account (or phone number) to be discovered, rather than forcing every user to have an account with a specific service or OSN.

Using OSNs for discovery and validation then opens the door for a generic key management platform that does not rely on trusted third parties to store private keys or to validate OSN accounts. A mobile application could be responsible for generating a keypair and posting, retrieving, and maintaining public keys on OSNs. The mobility of the application would allow the private key to be readily available, rather than stuck on a device that is not with you. In addition, the application could provide a crypto API that allows both local and remote (e.g. desktop) applications to encrypt, sign, verify, and decrypt arbitrary communications.

Two recent efforts provide some parts of this solution. Keybase provides a set of tools that allow users to generate PGP keypairs and post public keys to an OSN such as Twitter and Github, which implicitly verifies the authenticity of those keys to anyone who trusts those OSN accounts [4]. It also allows users to store a password-encrypted private key on the Keybase server for portability across devices. However Keybase falls short of providing a mobile application responsible for key management, automatic key discovery (by querying the services and OSN accounts associated with an identity), and automation of cryptographic operations. SafeSlinger provides a mobile application that automates key exchange among a group of users, but is primarily aimed at synchronous, in-person key exchange [2].

3. OPEN RESEARCH QUESTIONS

A wide range of open usability problems must be solved in order to provide social authentication:

- *Managing keys.* Users need methods for managing their public keys, including revoking keys when they are lost or stolen, and issuing new ones after expiration. Typical methods for coping with lost keys depend on a smartcard or a third party that can store private keys that are encrypted with a strong password. However, these methods introduce additional usability challenges, such as helping users to manage subkeys.
- *Inducting novices.* Our experience designing secure email systems indicates that novices need help transitioning to secure communication. Leveraging the users' OSN has the potential to ease the induction experience because it will include familiar identifiers and systems.
- *Authenticating strangers.* Authenticating people a user doesn't know well is a particular challenge. Perhaps a system should help people take different actions depending on the level of trust they have established. The web of trust has long been proposed as a way of helping determine the trustworthiness of someone who is known to your friends, but inferring trust is difficult [3] and little usability work has shed light on whether building a web of trust is viable.
- *Evaluating Resilience.* If someone's OSN account is compromised attackers may provide a fake public key for the compromised identity. The software will need to distinguish between this and regeneration of expired or lost keys (possibly by leveraging agreement among other OSN accounts for the compromised user). We also need to measure how vulnerable such a platform would be to Sybil and related attacks on OSNs that may provide inaccurate values of trust to users, directly or indirectly. Finally, standards are needed for measuring trust in both a user and a user's key.

We intend to develop a system that will allow us to evaluate solutions to these problems in both short and long term user studies.

4. REFERENCES

[1] W. Bai, D. Kim, M. Namara, Y. Qian, P. G. Kelley, and M. L. Mazurek. An inconvenient trust: User attitudes toward security and usability tradeoffs for key-directory encryption systems. In *Symposium On Usable Privacy and Security*, 2016.

[2] M. Farb, Y.-H. Lin, T. H.-J. Kim, J. McCune, and A. Perrig. Safeslinger: easy-to-use and secure public-key exchange. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 417–428. ACM, 2013.

[3] J. Golbeck and J. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology (TOIT)*, 6(4):497–529, 2006.

[4] Keybase. `https://keybase.io/`.

[5] S. Ruoti, J. Andersen, S. Heidbrink, M. O'Neill, E. Vaziripour, J. Wu, D. Zappala, and K. Seamons. "we're on the same page": A usability study of secure email using pairs of novice users. In *Proceedings of the 34th annual ACM conference on Human factors in computing systems, ACM*, 2016.

[6] S. Ruoti, J. Andersen, D. Zappala, and K. Seamons. Why Johnny still, still can't encrypt: Evaluating the usability of a modern PGP client. *arXiv preprint arXiv:1510.08555*, 2015.

[7] S. Sheng, L. Broderick, C. A. Koranda, and J. J. Hyland. Why Johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*, pages 3–4, 2006.

[8] A. Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Usenix Security*, volume 1999, 1999.

Design of Secure Chatting Application with End to End Encryption for Android Platform

Ammar Hammad Ali[1], Ali Makki Sagheer[2]

[1] College of Computer Science and Information Technology / University of Anbar
p.a.alfahad@gmail.com

[2] College of Computer Science and Information Technology / University of Anbar
ali.m.sagheer@gmail.com

Abstract: ***In this paper, a secure chatting application with end to end encryption for smart phones that used the android OS has been proposed. This is achieved by the use of public key cryptography techniques. The proposed application used the Elliptic Curve Diffie Hellman Key Exchange (ECDH) algorithm to generate the key pair and exchange to produce the shared key that will be used for the encryption of data by symmetric algorithms. The proposed Application allows the users to communicate via text messages, voice messages and photos. For the text message security the standard AES algorithm with a 128 bit key are used. The generated key (160 bit) minimized to 128 bit length by selecting the first 128 bit of the generated key in order to be used by the AES algorithm. For the voice and image security processes the proposed application used the symmetric algorithm RC4 for this purpose.***

Keywords: Android, Chatting Application, ECDH (Elliptic Curve Diffie Hellman Key Exchange), AES (Advanced Encryption Standard), RC4 (Rivest Cipher 4).

1. INTRODUCTION

The mobile instant message applications have overwhelmed the Short Message Service (SMS) operated by cellular network carriers, with 19 billion messages sent for every day contrasted and more than 17 billion SMS messages [1].

Instant message will assume an essential part later on business territories, which are prevalently known as m-commerce, mobile banking, administrative use, and everyday life correspondence. Moreover, instant message has turned into a famous wireless service all over the world as it encourages a client to be in contact with any mobile phone subscriber anyplace on the planet [2].

With the increasingly developing dependence on mobile chat system in one hand, and the developing number of vulnerabilities and assaults on the other hand, there is an undeniably interest for the security solutions. There are likewise some extra security issues in the wireless media that are not the situation in a wired framework. In this manner, extraordinary secure protocols are required for assortment mobile chat system platforms [3].

Customers utilize a mobile chat service to communicate with each other, a procedure that can incorporate relaying individual data. The security and protection of such communications ought to be considered important. In any case, late scenes of powerlessness in the significant chat services uncover that they won't be robustly actualizing security and protection highlights [4].

In the late years, Data Confidentiality, Authentication, Integrity, Non-repudiation, Access control, and Availability are the most imperative security services in the security criteria that ought to be considered in secure applications and frameworks. Notwithstanding, there is no arrangement for such security services in the mobile chat systems. Both mobile chat system customer and mobile chat system server are defenseless against both passive and active attacks. Passive dangers join arrival of message substance, and Traffic examination while active dangers consolidate adjustment of message substance, masquerade, replay, and denial of service (DoS). Truth be told, all the specified risks are appropriate to the mobile chatting communications [3].

The security and protection saving components of different versatile applications have gone under the spot-light. There are assorted security and protection highlights given by different mobile chat applications, yet there are not very many portable talk applications that give an End-to-End encryption administrations security to their customers [4].

2. RELATED WORKS

There are countless talk applications that claim to give a protected administration, however their total design is not freely accessible.

In 2013 Dec, Ali Makki Sagheer et al. proposed a solution that gives secrecy and uprightness to SMS data by applying a crossbred cryptographic plan which join the AES for encryption/unscrambling plan and RC4 for key extension and generation algorithms to satisfy all the more intense security issues. The proposed model is actualized by Java programming dialect in view of Net Beans platform. The proposed framework was tried on different cell phones, for example, the Nokia 5233.

Our work use Public Key Encryption algorithms that will save the time and cost spent to agree on a key between the users also the encryption time is minimized compered to this paper [5].

In 2014 May, H.C. Chen et al. [6] exhibited another idea about Mobile Text Chat utilizing a revolution session key based transposition cryptosystem plan. Their proposed conspire just manages the safe content transposition for mobile chat framework. It acclimatized the technologies of classical block cipher, substitution and transposition. Also, the new session key can be created by the network pivot innovation. It could be easily applied to transmit via mobile devices using the quick encryption algorithm.

Table 2: Text message encryption/decryption time

Size in Bytes	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
32	17	20	19	22	21	24
128	22	24	20	23	23	29
512	30	25	21	24	37	31
2048	34	27	23	26	39	33
4096	43	37	24	27	42	36

Table 3 shows the duration and the size of the tested voice messages, hence the max length of the voice message allowed in the proposed application is 60 Sec, and therefore, it is the max length tested.

Table 3: the voice message duration and size

NO	*Duration (Sec)*	*Size (KB)*
1	10	16
2	20	31
3	30	48
4	45	71
5	60	95

Table 4 shows the time of voice encryption and decryption processes in millisecond. The algorithm used for encrypting voice and image messages is the RC4 which is one of the fastest encryption techniques and it is suitable for the mobile device when encrypting vast amounts of data.

Table 4: voice message encryption/decryption time

No	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	3	2	2	2	3	1
2	7	4	4	4	5	2
3	11	7	7	5	6	3
4	15	11	9	8	10	5
5	29	20	13	10	16	6

Table 5 shows the examined image size, NPCR and UACI. The NPCR and UACI are intended to test the quantity of changing pixels and the quantity of averaged changed intensity between encrypted pictures.

Table 5: the image message size, NPCR and UACI

NO	*Size (KB)*	*NPCR*	*UACI*
1	26	99.59	33.986
2	66	99.62	29.135
3	118	99.61	32.694
4	181	99.60	29.887
5	220	99.62	32.616

The proposed application allows transfer images that have size less than 250 KB. So, the tested images have the allowed size only. Table 6 shows the time of images encryption and decryption processes in millisecond.

Table 6: Image message encryption/decryption time

No	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	89	74	53	47	124	51
2	163	182	107	103	132	102
3	296	291	168	164	155	161
4	420	399	248	242	171	124
5	463	424	261	257	213	149

9. CONCLUSION

In this paper, a secure chatting application was developed. The proposed application was tried on various mobile devices. According to the obtained results the following are summarized as conclusions.

End to End Encryption is achieved by involving ECDH key exchange to provide the key pair, which will be exchanged between the two parties to generate the secure shared key that will be used as a key for the encryption algorithms. The proposed secure chatting application furnish confidentiality, privacy and integrity. Users can be granted that nobody, even the provider of the service, cannot read their messages. The exchanged data is store only at the server, and nothing of them is stored at the physical memory of the phone. The algorithm used for encrypting text messages is the AES standard which is slower than other block cipher but it provides higher security. The algorithm used for encrypting voice and image messages is the RC4 which is one of the fastest encryption techniques and it is suitable for the mobile device when encrypting immeasurable sums of data.

REFERENCES

[1] Li Zhang, Chao Xu, Parth H. Pathak, and Prasant Mohapatra, "Characterizing Instant Messaging Apps on Smartphones", Passive and Active Measurement Lecture Notes in Computer Science, pp. 83-95, 2015.

[2] Medani1, A. Gani1, O. Zakaria, A. A. Zaidan, and B. B. Zaidan, "Review of mobile short message service security issues and techniques towards the solution", Scientific Research and Essays Vol. 6(6), pp. 1148-1165, March 2011.

[3] Hsing-Chung Chen, Jyh-Horng Wen and Cheng-Ying Yang, "A Secure End-to-End Mobile Chat Scheme", Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, 2014.

[4] Raja Naeem Akram, and Ryan K. L. Ko, "End-to-End Secure and Privacy Preserving Mobile Chat Application", Information Security Theory and Practice. Securing the Internet of Things Lecture Notes in Computer Science, pp.124-139, 2014.

[5] Ali Makki Sagheer, Ayoob Abdulmunem Abdulhameed and Mohammed Adeeb AbdulJabbar, "SMS Security for Smartphone", Sixth International Conference on Developments in eSystems Engineering, 2013.

[6] H.C. Chen and A.L.V. Epa, "A Rotation Session Key-Based Transposition Cryptosystem Scheme Applied to Mobile Text Chatting", Proceedings of The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA2014), pp. 497 - 503, Victoria, Canada, May 2014.

[7] Pejman Dashtinejad," Security System for Mobile Messaging Applications ", Thesis, KTH University, Jan 2015.

[8] S. Kumar, M. Girimondo, A. Weimerskirch, C. Paar, A. Patel, and A. S. Wander, "Embedded End-to-End Wireless Security with ECDH Key Exchange", 2003 46th Midwest Symposium on Circuits and Systems.

Scientific Research Publishing

Journal of Information Security, 2018, 9, 95-99
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger

Robert E. Endeley

Capitol Technology University, Laurel, MD, USA
Email: reendeley@captechu.edu

How to cite this paper: Endeley, R.E. (2018) End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger. *Journal of Information Security*, **9**, 95-99.
https://doi.org/10.4236/jis.2018.91008

Received: December 22, 2017
Accepted: January 20, 2018
Published: January 23, 2018

Abstract

The ubiquity of instant messaging services on mobile devices and their use of end-to-end encryption in safeguarding the privacy of their users have become a concern for some governments. WhatsApp messaging service has emerged as the most popular messaging app on mobile devices today. It uses end-to-end encryption which makes government and secret services efforts to combat organized crime, terrorists, and child pornographers technically impossible. Governments would like a "backdoor" into such apps, to use in accessing messages and have emphasized that they will only use the "backdoor" if there is a credible threat to national security. Users of WhatsApp have however, argued against a "backdoor"; they claim a "backdoor" would not only be an infringement of their privacy, but that hackers could also take advantage of it. In light of this security and privacy conflict between the end users of WhatsApp and government's need to access messages in order to thwart potential terror attacks, this paper presents the advantages of maintaining E2EE in WhatsApp and why governments should not be allowed a "backdoor" to access users' messages. This research presents the benefits encryption has on consumer security and privacy, and also on the challenges it poses to public safety and national security.

Keywords

Instant Messaging, WhatsApp, End-to-End Encryption, National Security, Privacy

1. Introduction

The world is ever changing due to the advancement in the realm of science and technology, and these days it seems hard to escape the presence of technology in

question." [9]. WhatsApp Inc. has since responded to this claim, saying that the feature in question is a design tradeoff, meant to prevent users from losing their messages if they switch phones or reinstall the app.

3. Conclusion

While a majority of countries would favor some kind of restriction on access to unrecoverable encryption, there is no global consensus, and the likely outcome is a hodgepodge of national policies. According to [10], "Our research suggests that the risk to public safety created by encryption has not reached the level that justifies restrictions or design mandates". Lewis *et al.* further went on to say, "The encryption issue that law enforcement faces, while frustrating, is currently manageable". Communications privacy is a key element of human rights in the digital era, and developments affecting it ought to be reported. Ultimately, removing WhatsApp E2EE would not be the solution, as criminals could create their own, similar software that would allow them to communicate securely, while ordinary users would lose the ability to send genuinely private messages [6]. Maintaining E2EE in WhatsApp without an encryption backdoor guarantees genuine privacy in conversations between individuals and group chats. Voice conversations through WhatsApp messenger feel more natural; users are assured that no one is eavesdropping on their conversations, and conversations thus tend to feels more like a face-to-face conversation.

References

[1] Yeboah, J. and Ewur, G. (2014) The Impact of WhatsApp Messenger Usage on Students Performance in Tertiary Institutions in Ghana. *Journal of Education and Practice*, **5**, 157-164.

[2] Sarker, G.R. (2015) Impact of WhatsApp Messenger on the University Level Students: A Sociological Study. *International Journal of Natural and Social Sciences*, **2**, 118-125.

[3] Jisha, K. and Jebakumar (2014) A Trend Setter in Mobile Communication among Chennai Youth. *IOSR Journal of Humanities and Social Science* (*IOSR-JHSS*), **19**, 01-06.

[4] Central Intelligence Agency (2017) The World Factbook. Country Comparison, Internet Users. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html

[5] Pascual, A. (2013) Data Breaches Becoming a Treasure Trove for Fraudsters, 2013 Identity Fraud Report. https://www.javelinstrategy.com/coverage-area/2013-identity-fraud-report-data-breaches-becoming-treasure-trove-fraudsters

[6] Michalas, A. (2017) How WhatsApp Encryption Works—And Why There Shouldn't Be a Backdoor. The Conversation. https://theconversation.com/how-whatsapp-encryption-works-and-why-there-shouldnt-be-a-backdoor-75266

[7] District Attorney New York County (2005) Going Dar: Encryption, Technology and the Balance between Public Safety and Privacy. District Attorney New York County,

An Encryption Protocol for End-to-end Secure Transmission of SMS

Minta Thomas
PG Student,CSE
Toc-H Institute of Science and Technology
Ernakulam,India.Pin-682313
Email:mintathomasj7@gmail.com

Panchami V
Assistant Professor ,CSE
Toc-H Institute of Science and Technology
Ernakulam,India.Pin-682313
Email: panchamam036@gmail.com

***Abstract*-Short Message Service (SMS) is a process of transmission of short messages over the network. SMS is used in daily life applications including mobile commerce, mobile banking, and so on. It is a robust communication channel to transmit information. SMS pursue a store and forward way of transmitting messages. The private information like passwords, account number, passport number, and license number are also send through message. The traditional messaging service does not provide security to the message since the information contained in the SMS transmits as plain text from one mobile phone to other. This paper explains an efficient encryption protocol for securely transmitting the confidential SMS from one mobile user to other which serves the cryptographic goals like confidentiality, authentication and integrity to the messages. The Blowfish encryption algorithm gives confidentiality to the message, the EasySMS protocol is used to gain authentication and MD5 hashing algorithm helps to achieve integrity of the messages. Blowfish algorithm utilizes only less battery power when compared to other encryption algorithms. The protocol prevents various attacks, including SMS disclosure, replay attack, man-in-the middle attack and over the air modification.**

Index terms: Cryptography, Encryption, Secure Transmission, Symmetric Encryption, Asymmetric Encryption.

I. INTRODUCTION

Mobile phones have become pervasive and ubiquitous in the current environment around the world. Short Message Service (SMS) or text messaging is one of the services that have been very popular in the mobile phones. SMS has become one of the fastest and strong communication channels to transmit the information. It is crucial to protect the content of the message when confidential information is exchanged using SMS. The data can be protected using cryptography.

"Cryptography" derives from the Greek word kruptos which means "hidden". Cryptography is an area of computer science which is developed to provide security for the senders and receivers to transmit and receive confidential data through an insecure channel. Cryptography [1] ensures that the message should be sent without any alterations and only the authorized person can be able to open and read the message.

The plain text is the original message that a person wishes to send to other user. The Cipher text is the message that cannot be understood by others. A key can be a numeric or alpha numeric text or a special symbol. Encryption is the process of converting the plaintext into cipher text using a key. Decryption is the reverse process of encryption in which the original message is retrieved from the cipher text.

The encryption and decryption process is given in Fig. 1. The original message or the plain text is the input to the encryption process which encrypts the plaintext using a key and produces a cipher text to be transmitted. The input cipher text is passed through the decryption process which decrypts the cipher text using the same key as that of encryption at the decryption end. Finally the original plaintext message is obtained.

Fig. 1: Encryption and Decryption Process

A. Goals of cryptography

Cryptography provides some security services that ensure adequate security of the systems or of data transfers. Following are the main cryptographic goals [2]:

978-1-4799-7075-9/15/$31.00©2015 IEEE

V. IMPLEMENTATION

The proposed work aims to achieve the cryptographic goals by making use of protocol and algorithms as discussed in proposed work. The overall system can be divided into three modules, namely, User Profile, Authentication Server and Secure Communication.

A. User Profile

When a mobile user wants to send a confidential information to another user, the user requests for a connection to the authentication server. The request includes the MAC or timestamp of the message. The authentication server forwards the requests to certified authority where all the mobile subscriber details are stored. If it is a valid user, then the user is connected to server.

B. Authentication Server

An authentication server (AS) stores the keys shared between the authentication server and the users. On conforming the connection between the sender and receiver the authentication server sends the secret key to both users. The encryption takes place through this secret key.

C. Secure Communication

The sender calculates the hash value of the message using MD5 and encrypts the message using the Blowfish algorithm. The key for encryption is send by the authentication server. On receiving the message the receiver decrypts the ciphered message and calculates the hash. The error detection is performed by comparing the received and calculated hash.

VI. CONCLUSION

The SMS are being used in many daily life applications. But when we send an SMS from one mobile phone to other, the information contained in the SMS transmit as plain text. The paper explains an efficient encryption protocol which includes the Blowfish and MD5 algorithms that aims to achieve confidentiality and integrity respectively. The EasySMS protocol provides authentication. The proposed work aims to achieve high throughput, encrypts more faster and saves battery power as it makes use of energy efficient Blowfish algorithm. Hence the SMS can be securely transmitted from one mobile to another.

REFERENCES

[1] Monika Agrawal, A Comparative Survey on Symmetric Key Encryption Techniques, International Journal on Computer Science and Engineering (IJCSE),Vol. 4 No. 05 May 2012.

[2] William Stallings, Cryptography and Network Security Principles and Practices Fourth Edition.

[3] Ketu File white papers, Symmetric vs Asymmetric Encryption, a division of Midwest Research Corporation.

[4] M. Hassinen, Java based public key infrastructure for SMS messaging,in Proc. 2nd ICTTA, 2006, pp. 8893.

[5] S. Wu and C. Tan, A high security framework for SMS,in Proc. 2nd Int. Conf. BMEI, 2009, pp. 16.

[6] A. De Santis,G. Cattaneo, M. Cembalo, F. Petagna, and U. F. Petrillo, *An extensible framework for efficient secure SMS*,in Proc. Int. Conf. CISIS, 2010, pp. 843850.

[7] J. L.-C. Lo, J. Bishop, and J. H. P. Eloff, *SMSSec: An end-toend protocol for secure SMS*, Computer Security, vol. 27, nos. 56,pp. 154167, 2008.

[8] M. Toorani and A. Shirazi, *SSMS: A secure SMS messaging protocol for the m-payment systems*,in Proc. IEEE ISCC, Jul. 2008, pp. 700705.

[9] Tingyuan Nie, Chuanwang Song, Xulong Zhi, *Performance Evaluation of DES and Blowfish Algorithms*,IEEE, 2010.

[10] Diaa Salama Abdul Minaam, Hatem M. Abdual-Kader, and Mohiy Mohamed Hadhoud, *Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types*,International Journal of Network Security, Vol.11, No.2, PP.7887, Sept. 2010.

[11] Najib A. Kofahi, Turki Al-Somani and Khalid Ai-Zamil, *Performance Evaluation of Three Encryption/Decryption Algorithms*,IEEE,2010.

[12] Diaa Salama,Hatem Abdual Kader, and Mohiy Hadhoud, *Wireless Network Security Still Has no Clothes*,International Journal on Computer Science and Engineering (IJCSE),Vol. 4 No. 05 May 2012.

[13] P. Ruangchaijatupon, P. Krishnamurthy, *Encryption and Power Consumption in Wireless LANs-N*, The Third IEEE Workshop on Wireless LANs - September 27-28, 2001- Newton, Massachusetts.

[14] Pratap Chnadra Mandal, *Superiority of Blowfish Algorithm*,International Journal of Advenced Research in Computer Science and Software Engineering 2(9),September - 2012, pp. 196-201.

[15] R. Rivest, *The MD5 Message-Digest Algorithm*,Network Working Group, 1992.

[16] Zhang Qing, *Iterative Hashing Algorithm Base on MD5*, Journalon Computer Engineering,vol.37(18) 124-126,2011.

[17] A. A. Pamungkas, *Implementasi Algoritma Sistem Kriptografi MD5,SHA-1 dan RC4 pada Aplikasi Mobile Internet Berbasis Java*,Journal Penelitian dan Pengembangan Telekomunikasi, vol. 11, no. 1, June 2006.

END TO END ENCRYPTION: AN ANSWER TO SECURITY CONCERNS IN THE PRIVATE SECTOR

Erik Wehner, edw34@pitt.edu, Lora, 3:00, Evan Moran, ecm61@pitt.edu, Lora, 1:00

***Abstract*—Personal online security is becoming an extremely important topic in engineering. This is due to the increased popularity of online communication. The best technology to ensure a user's data is safe is end to end encryption. Ever since electronic mailing services became available to the public, questions arose about how secure these messages were. In addition, the popularity of online shopping, banking, and other financial online transactions brought forth an even higher demand for the security of the internet. An example of technology that can provide a high level of privacy is end to end encryption. Hardware embedded into phones and computers allows for the random locks and keys that make up end to end encryption only work on the devices involved in the conversation. This allows for the data that is being transmitted to be completely secure. End to End encryption is very useful for the sustainability of privacy online, as it protects users from attacks and it is very hard to crack. iMessage is the best example of this extremely important technology in action on a large scale. The end to end encryption of the messages makes it almost impossible to intercept an iMessage message. In addition, the history of encryption as it relates to this software being used by the public is important to the understanding of how end to end encryption evolved into how it is known today in applications like iMessage. Apple's iMessage software is one of the most popular publicly available messaging software in use, therefore serving as a representative of the effectiveness of end to end encryption technology. Although this technology allows extremely high levels of security for users, this comes with some concerns, such as the limits encryption places on government surveillance.*

Key Words- AES, Cryptography, Cybersecurity, Encryption, End to End, Internet Privacy, Private Messaging

THE IMPORTANCE OF ONLINE SECURITY

As more and more people use the internet as a means to communicate, the importance of securing people's and business' online communications becomes more imperative and a much more difficult engineering challenge. According to the Central Intelligence Agency, there were an estimated 276 million internet users in the United States in 2014, and that number is predicted to rise [1]. With this many users, the incentive for hackers to execute attacks increases. CNET reported that in September of 2016, Yahoo! confirmed that 500 million accounts were hacked into, making it the largest data breach in history. Through this massive number of email accounts, names, other email addresses, contacts, phone numbers, security question answers, and contacts were all stolen. With this information, hackers had the ability to send out emails in the victim's name and also read through all of the emails associated with the account. A majority of these hacked accounts are still being sold on the internet to people with the intent of stealing personal information, hacking into other accounts, and possibly even using the email to hack into an associated bank account [2]. According to a Javelin Strategy and Research Report in 2012, one in every four people that have a breach in their online data become a victim of identity theft as a result of that. [3]. This reveals the connection between online data breaches and hackers using this information to steal a victim's identity. End to end encryption provides an effective way to prevent against these attacks through technology that allows the algorithms to be almost unbreakable. This technology, if implemented properly, could have prevented this large-scale attack. End to end encryption offers a sustainable solution to the ongoing problem of online security. According to *Urban Sustainability in Theory and Practice*, sustainability is referred to as intersecting "with other social conditions, such as resilience, livability, adaptation, innovation and reconciliation, as basic conditions of positive social life." [4]. This technology offers peace of mind to users on the internet that their data is safe. In addition, end to end encryption is a technology that will remain relevant and in use for a long period of time, due to its security and adaptability. Overall, online security is very important in today's society, however the concept of encryption has been around for centuries.

THE HISTORY OF ENCRYPTION

Earliest Forms of Encryption

The idea of encrypting information so it is not available to be understood by anyone but the ended recipient of a message is not a new idea. An article published by the University of Utah reports that the first known example of cryptography dates back to an Egyptian town in 1900 BCE. A series of hieroglyphics with a number of unusual symbols was

organizations. This is commonly looked down upon by consumers, because they are not getting a service that is truly encrypted end to end.

A very common example of an end to end encryption service with a backdoor as a part of the system is Microsoft's Skype. The service was previously thought to be fully end to end encrypted without a way for Microsoft to see what is sent. In 2013, Edward Snowden shared that the platform did in fact have back door, in which enabled Microsoft access to all messages and communications on the service. Microsoft and Skype denied that statement many times, but in the end, it was shown that Skype messaging and voice backdoors. [22]. Skype users protested the system, because it allowed attackers an easier way to access anybody's information, because Microsoft has already decrypted their own technology, which means that it is easier to decrypt it.

There is an ethical dilemma that Microsoft faced when they had to implement backdoors. By implementing back doors, it meant that the service was not truly end to end encrypted, which meant that it was as not as secure as it was thought out to be. In order to protect the general public, the backdoor was implemented, which meant that Microsoft and Skype had to protect the public. Usually, Microsoft would not really need to access messages sent, for most people do not have ill intentions. It can be understood, that after the fact, Microsoft and Skype wants to be able to access records to provide the government and any government agencies with information that is able to protect the public in the future.

FUTURE OF ENCRYPTION

Encryption and end to end encryption has long been a great tool that has helped us many times, and the future of end to end encryption as a tool in the public really depends on the decisions of today and soon to come. If end to end encryption is accepted generally as the normal, then there is a wild amount of possibilities open in the future for encryption and cryptography. In 2010, and Idea was put forth for a type of encryption called fully homomorphic encryption, which allows access to data without ever decrypting it. Another idea is called honey encryption, in which upon many wrong guesses of a key creates data that is not accurate but looks like it is. Next is called functional encryption, in which restricted keys enable the key holder to learn only about a function of data and nothing else. The final approach is called quantum key encryption, in which the quantum nature of atoms actually is responsible for protecting all of the user's data [23]. These methods could be even more secure than end to end encryption, which can help keep people's information safe, in both the communication platforms and patient data. Data of a person could potentially be saved on that persons atoms and cells, which could potentially be very safe, because only the person could decrypt the information on their atoms, plus the have access to their atoms. Encryption is a very safe way to protect a user's confidential data, and is essential in this age, an age that is filled with technology and progressive, creative thinkers who will further accomplishment and achievement.

SOURCES

[1] "Country Comparison." Central Intelligence Agency. 01.01.2014. Accessed 02.22.2017. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html

[2] L. Hautala. "Yahoo hit in worst hack ever, 500 million accounts swiped" CNET. 9.22.2016. Accessed 10.28.2016. https://www.cnet.com/news/yahoo-500-million-accounts-hacked-data-breach/

[3] N. Ozawa "More Than 12 Million Identity Fraud Victims in 2012." Javelin Strategy. 2.20.2013. Accessed 10.28.2016 https://www.javelinstrategy.com/press-release/more-12-million-identity-fraud-victims-2012-according-latest-javelin-strategy-research

[4] James, Paul, Liam Magee, Andy Scerri, and Manfred Steger. *Urban Sustainability in Theory and Practice: Circles of Sustainability*. Abingdon (Oxon): Routledge, 2015. Print.

[5] N. McDonald. "Past, Present, and Future Methods of Cryptography and Data Encryption." University of Utah. 5.09.2009. Accessed 2.25.2017. http://www.eng.utah.edu/~nmcdonal/Tutorials/EncryptionResearchReview.pdf

[6] "The Zimmermann Telegram." Cryptologic Quarterly. Vol. 20 no 2. 06.23.2001. pp. 43-55

[7] M. Cozzens, S. Miller, et al. "The Mathematics of Encryption: An Elementary Introduction." Rutgers University. Accessed 02.23.2017.

[8] D. Mowry. "German Cipher Machines of World War II." National Security Administration. 03.24.2014 pp 3.

[9] "The Enigma of Alan Turing." Central Intelligence Agency. 04.10.2015. Accessed 02.26.2017. https://www.cia.gov/news-information/featured-story-archive/2015-featured-story-archive/the-enigma-of-alan-turing.html

[10] M. Behringer "End-to-End Security" The Internet Protocol Journal. Vol. 12 no 3. pp 1.

[11] T. Dierks, E. Rescorla. "The Transport Layer Security (TLS) Protocol." Internet Engineering Task Force. 08.01.2008. Accessed 02.15.2017. https://www.ietf.org/rfc/rfc5246.txt

[12] W. Diffie, M. Hellman. "New Directions in Cryptography." IEEE Transactions on Information Theory. Vol. 22 no 6. 11.03.1976

[13] D. Van Steen. "Strategies for Securing Distributed Systems." Pace University. Accessed 02.22.2017. http://csis.pace.edu/~marchese/CS865/Lectures/Chap9/Chap9New/Chapter9_html_5c91172f.jpg

[14] D. Adrian, K Bhargavan, Z. Durumeric, et al. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice." Proceedings ACM SIGSAC. 2015. pp. 5-17

[15] "Announcing the Advanced Encryption Standard."