

#### Article Citation Format

Sajoh, D.I., Atiku, A.U. & Naibi, R.S. (2018): Secure Messaging: Analysis of Signal Protocol Implementation in WhatsApp and Signal Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 6, No. 3. Pp 63-72

#### Article Progress Time Stamps

Article Type: Research Article  
 Manuscript Received: 2<sup>nd</sup> Aug, 2018  
 Review Type: Blind  
 Final Acceptance: 18<sup>th</sup> Aug, 2018  
 Article DOI: [dx.doi.org/10.22624/AIMS/DIGITAL/V6N3P6](https://doi.org/10.22624/AIMS/DIGITAL/V6N3P6)

## Secure Messaging: Analysis of Signal Protocol Implementation in WhatsApp and Signal

Sajoh, D.I., Atiku, A.U. & Naibi, R.S.

Modibbo Adama University of Technology Yola

Email ids: <sup>1</sup>disajoh@mautech.edu.ng, <sup>2</sup>ahmed.atiku@mautech.edu.ng, <sup>3</sup>ridwan.naibi@mautech.edu.ng

### ABSTRACT

The movement of different messaging applications towards secure messaging has made it crucial to analyze the different properties that these applications should achieve. In this work, the different properties of secure messaging, with regards to trust establishment, conversation security, and transport security; were analysed. Two messaging applications, WhatsApp and Signal, that are using the same protocol for end-to-end encryption were compared based on how they satisfied those properties. The areas where the two applications are similar and where they differ were discussed. Some trade-off made by the WhatsApp and Signal in their implementation of the signal protocol were highlighted.

**Keywords**—Secure Messaging, WhatsApp, Signal, Security, Usability

### 1. INTRODUCTION

WhatsApp is a popular messaging application with over a billion users and it supports end-to-end encryption [1]. The end to end encryption in WhatsApp is based on the Signal Protocol designed by Open Whisper Systems. Open Whisper Systems also owns a messaging application called Signal which also uses the Signal Protocol[3]. There is a concern of mass surveillance on messaging applications[9][7]. This makes it important to analyse the cryptography of secure messaging applications. Though WhatsApp and Signal use the same protocol for end-to-end encryption, there are a few differences in the implementation. Some of these differences lead to a polemic against WhatsApp[5]. This makes it necessary to analyse the differences.

This work analysed the similarities and differences between the end to end encryption in WhatsApp and Signal and discussed some of the issues surrounding WhatsApp's implementation of the Signal Protocol.

### 2. SECURE MESSAGING

Secure messaging applications achieve end-to-end encryption by addressing three major problem areas: Trust Establishment, Conversation Security and Transport Privacy[11]. In each of these problem areas, there are some properties which the applications are expected to achieve. These properties are categorized based on Security and Usability. Due to technology limitations and/or conflicting requirements, not all these properties can be achieved, hence tradeoffs become necessary. In the sections that follows, we discussed the Security and Usability properties of the Messaging in Trust

Establishment; the properties of Conversation Security and Transport Privacy respectively.

## 2.1 Trust Establishment

Trust establishment is concerned with the challenges of ensuring that parties in a communication are using the correct keys for encrypting and decrypting their messages and the process of distributing these keys does not leak any information to a third party, who is not part of the communication.

### 2.1.1 Security Properties

The desired security properties for Trust Establishment are as follows:

- **Prevent Man in the Middle Attacks (MITM):** This is a very important security property for trust establishment. Secure messaging solutions are supposed to be designed in such a way that even the operator cannot read the conversation between two users. Therefore, it is important that the solution does not give room for the Operator or any other third party to perform MITM attacks.
- **Operator Accountability:** If the solution to some extent provides protection against MITM attacks, the user should be able to check and make sure that the operator is following the protocol.
- **Key Revocation Allowed:** A user should be able to change their cryptographic keys. This is important if the user notices that their keys have been compromised.
- **Preserves User Privacy:** The meta-data should not be read by the operator or any third party.
- **Detect Key Change:** Users should be able to detect when the cryptographic keys were changed.
- **Prevent use of unverified keys:** A user should be able to stop the application from encrypting messages with keys he/she has not verified.

The properties "Detect Key Change" and "Prevent using unverified keys" were not in the analysis in [11]. They were added because they are relevant to the analysis in section 3.

### 2.1.2 Usability Properties

No matter how secure an application is, if it is not usable, it will not be adopted by users. Usability properties are therefore as important as the security properties. The desired usability properties of trust establishment are as follows:

- **Automatic Key Generation:** The application should be able to generate keys without user effort.
- **No Key Management:** Users should not have to engage in key management.
- **In-band:** There should be no external means of ensuring security of trust establishment.
- **Alert-less Key Renewal:** When a user changes his/her key, other users communicating with him/her should not be disturbed with warning messages.
- **Inattentive User Resistant:** Users should not be expected to have an active role in ensuring the security of the trust establishment.
- **Asynchronous:** It should not be necessary for both users to be on-line to achieve trust establishment.
- **Scalability:** Trust establishment should not require a lot of additional resources if number of participants increase.
- **Auditing not Required:** No external auditors required in verifying that operators behave correctly.
- **Non-blocking:** Change in user keys should not interfere with the communication flow between users. This property is not in [11].

### 3. CONVERSATION SECURITY

Once trust establishment is achieved, the next thing is making sure that encryption and decryption of messages are done securely and without usability problems. To do that the following security and usability properties as described in [11] are desirable.

#### 3.1 Security Properties

The desirable security properties for conversation security are:

- **Confidentiality:** Messages sent by users should only be decrypted by intended recipients.
- **Integrity:** Messages that are altered by a third party should be detected and rejected by recipients.
- **Authentication:** Users should be able to verify the source of messages they received.
- **Participant Consistency:** All participants in the conversation should maintain the same list of participants at all times.
- **Destination Validation:** Users should be able to verify that the messages they receive was actually intended for them.
- **Forward Secrecy:** When the cryptographic keys are compromised by an attacker, the attacker should still fail to decrypt previous conversation.
- **Backward Secrecy:** When the cryptographic keys are compromised the attacker should fail to decrypt future messages.
- **Anonymity Preserving:** Conversation security should not leak any user information hidden through transport privacy.
- **Global Transcript:** All users should have same copy of conversation in the correct order.
- **Message Repudiation:** Given the communication transcript, an external judge should not get cryptographic prove that a message is sent from a certain user.
- **Participant Repudiation:** There should be no cryptographic prove that a user is part of a conversation from the communication transcript even if all but the user's cryptographic keys are available.
- **Message Unlinkability:** If an external judge is able to determine that a certain message is from a user, that should not be an evidence to show that another message is from the user.
- **Open Source:** The protocol should be open to external review
- **Computational Equality:** In a group chat, no user should have more computational load than another user.
- **Trust Equality:** All users should be equally trusted in the conversation in a group chat.
- **Subgroup Messaging:** In a group chat, a user should be allowed to send a message to a subset of the group members without the need of creating a new group.
- **Contractable Membership:** If a user leaves a group, there should be no need to restart the protocol and that user should not have the ability to decrypt messages in the group chat after leaving the group.
- **Expandable Membership:** If a user joins a group, there should be no need to restart the protocol and that user should not have ability to decrypt messages that were in the group chat before he/she joined.

The Open Source property is not included in [11], but was added because it is important in the analysis in the comparison between WhatsApp and Signal in section 3.

### 3.1.1 Usability Properties

The desirable usability properties of conversation security are:

- **Out of Order Resilient:** Delays caused by network or other factors might cause messages to arrive late or out of order. If that happens, the protocol should handle it without making any message inaccessible.
- **Asynchronous:** A user should be able to send messages to another user even if the second user is off-line. When a user gets on-line, the user should see all the messages sent to him/her while he/she was off-line.
- **Multi-Device Support:** A user should have the ability to use multiple devices for one account.
- **No Additional Service:** Conversation security should be achieved without the need for additional infrastructure.

## 4. TRANSPORT PRIVACY

Message Privacy aims at hiding the meta-data of the users in the transport layer making them anonymous.

### 4.1 Security Properties

The desired security properties of Transport Privacy are:

- **Sender Anonymity:** When a message is received, only the sender of a message should know who sent the message.
- **Recipient Anonymity:** When a message is sent, only the recipient should know he/she received the message.
- **Participation Anonymity:** Only the users that are part of a conversation should determine the network nodes used in the conversation.
- **Unlinkability:** Given two messages, only the users that are part of a conversation should determine if the messages are part of the same conversation or not.
- **Global Adversary Resistant:** The service providers or government entities that control a large portion of a network should not break the anonymity of the protocol even if they want to.

### 4.2 Usability Properties

The desired Usability properties of Transport Privacy are:

- **Contact Discovery:** If a user wants to send a message to another user, the system should provide a mechanism for getting the contact details.
- **No Message Delays:** Transport privacy should not cause unnecessary delays in sending and receiving messages.
- **No Message Drops:** It should not cause failure in retransmitting messages.
- **Easy Initialization:** Users should find it easy to start communication.
- **No Fees Required:** There should be no financial cost of using the system.
- **Topology Independent:** Transport privacy should not depend on any network topology.
- **No additional Service:** additional infrastructure should not be required.
- **Flood Resistant:** It should be resistant to denial of service attacks.
- **Low Storage Consumption:** Transport privacy properties should not use large storage space.
- **Low Bandwidth:** Transport privacy properties should not require a large bandwidth.
- **Low Computation:** It should not require a lot of processing power to achieve transport privacy.

- Asynchronous: A user should not have to be on-line to be sent a message. He/she should see the message when he/she gets on-line.
- Scalable: If number of users increase, there should not be a very large increase in the amount of required resources.

## 5. WHATSAPP VS SIGNAL

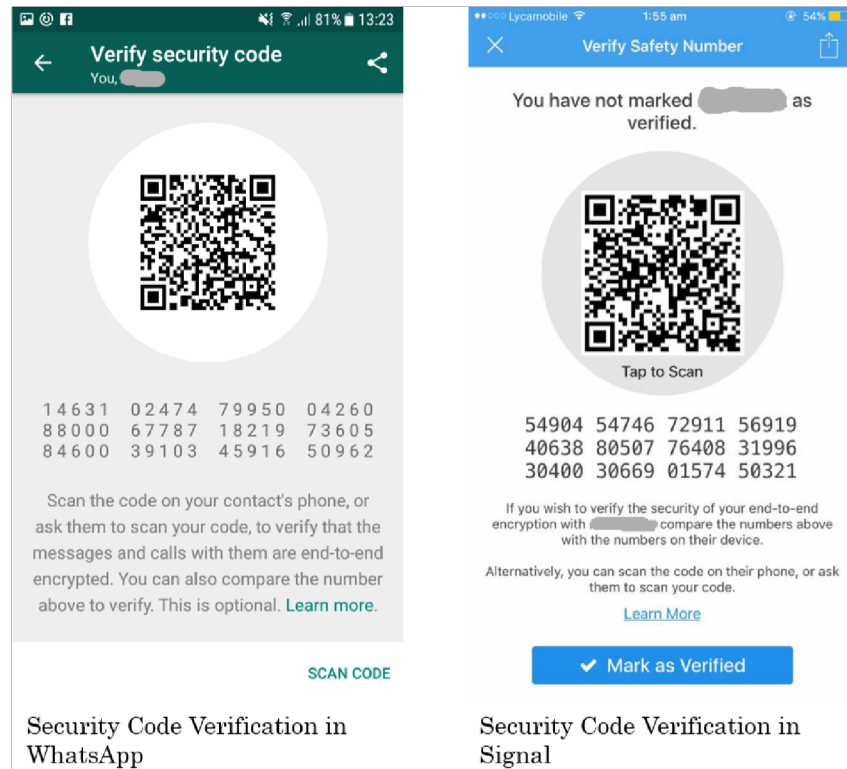
This section compares the implementation of the signal protocol in WhatsApp and Signal and identifies the desired properties discussed in section 2 that they satisfy. Section 3.1 discusses the similarities between WhatsApp and Signal with regards to how they satisfy secure messaging properties. The differences between them are discussed in section 3.2 and section 3.3 discusses a WhatsApp polemic.

### 5.1 Similarities

Since both WhatsApp and Signal are using the Signal Protocol, it is expected that they will have a lot of similarities in the properties they satisfy. In trust establishment, they both partially satisfy the **prevent MITM** and **Operator Accountability** properties. This is because The Signal Protocol is based on Trust-On-First-Use (TOFU) scheme and it does not prevent MITM at install time [11]. Both Signal and WhatsApp have added additional security at registration where users are sent verification code through Short Message Service (SMS) to the phone number they are registering. This additional layer still does not fully prevent MITM since SMS is insecure.

The key generation and management in both applications happen in the background and do not need user interaction [11][12]. This makes it lose the **Key Revocation Allowed property** and satisfy the **Automatic Key Generation** and **No key Management** properties[11]. This is a good example of a trade-off between security and usability. In this case, both applications sacrifice a security property to gain usability.

They both have a functionality for verifying users cryptographic keys which enable users to compare their security codes manually [12][3]. Figure 1 shows the security codes. This verification is optional, so it does not stop them from satisfying the **In-Band** property. If a user decides not to check the security code, the user cannot detect if there is a MITM attack by the operator in both applications making them loose the **Inattentive User Resistant** property.



**Figure 1: The security code used in verifying contact's security details in WhatsApp and Signal**



In both WhatsApp and Signal, a user does not have to be online for other users to find their cryptographic keys [11]. Both applications also require no external auditing and are scalable thereby satisfying the **Auditing Not Required** and **Scalability** properties. Table 1 gives a summary of the Trust establishment properties satisfied by WhatsApp and Signal. When it comes to conversation Security, most of the security properties are satisfied by the Signal Protocol. Both Signal and WhatsApp are not anonymous messaging applications and do not satisfy the **Anonymity Preserving** property [11]. Both applications do not have a global communication transcript [11]. Table 2 shows the summary of the Conversation Security properties satisfied by both WhatsApp and Signal. Since anonymity is not a major concern in the two applications, Transport privacy properties will not be discussed.

## 5.2 Differences

There are few differences in the implementation of the Signal Protocol by WhatsApp and Signal. This causes few differences in the properties that they satisfy. In Conversation Security, the only difference between the two is that Signal is open-source and WhatsApp is not. The other differences are from the Trust Establishment properties. The difference in trust establishment is from how the two applications handle key change. In Signal, when a user's cryptographic keys change, other users communicating with him/her will be notified. WhatsApp by default does not do that, but it has the option of enabling it. This makes WhatsApp satisfy **Alert-less Key Renewal** and Signal does not. On the other hand, it makes WhatsApp have the possibility of not **detecting key change** which makes it fail in satisfying that property while signal fully satisfies it. Another difference is that Signal stops retransmission of messages pending user's approval if keys are changed while WhatsApp does not as shown in figure 2.

This makes Signal partially **prevent the use of unverified keys** and WhatsApp completely fail to satisfy that property. On the other hand, it makes WhatsApp **Non-Blocking** and Signal blocking.

**Table 1: Comparison of Trust Establishment in WhatsApp and Signal**

Trust Establishment		 WhatsApp	 Signal
Security Properties	Prevent MITM	●	●
	Operator Accountability	●	●
	Key Revocation Allowed	✗	✗
	Preserves User Privacy	✓	✓
	Detect Key Change	●	✓
	Prevent Use of Unverified Keys	✗	●
Usability Properties	Automatic Key Generation	✓	✓
	No Key Management	✓	✓
	In-Band	✓	✓
	Alertless Key Renewal	✓	✗
	Inattentive User Resistant	✗	✗
	Asynchronous	✓	✓
	Scalability	✓	✓
	Auditing Not Required	✓	✓
	Non Blocking	✓	✗

✓ = Fully Satisfied

● = Partially Satisfied

✗ = Not satisfied

### 5.3 WhatsApp Polemic

The non-blocking nature of WhatsApp raised some criticisms on its end-to-end encryption. WhatsApp was accused of having a retransmission vulnerability in 2016 [8]. The claim is that: If Alice sends a message to Bob and the message is left in transit, probably because Bob is offline. An attacker can take advantage of the fact that SMS is not secure to register Bob's number with the server. Since WhatsApp is non-blocking and it does not stop retransmission of messages when keys are changed as shown in section 3.2, the message will be automatically re-transmitted and the attacker will be the one to receive it. This led to a publication on the Guardian with the title *"WhatsApp back door allows snooping on encrypted messages"* [5] which is a strong accusation. Open Whisper Systems later explained that it was a trade-off between security and usability as explained in section 3.2. Open Whisper Systems also mentioned that the attack mentioned is difficult in practice [4]. After facing criticisms [2], the Guardian publication was edited to the one shown in figure 3 [6].

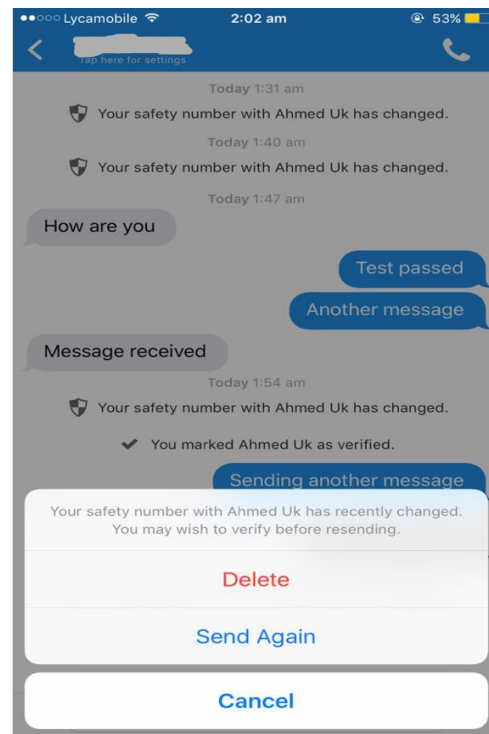
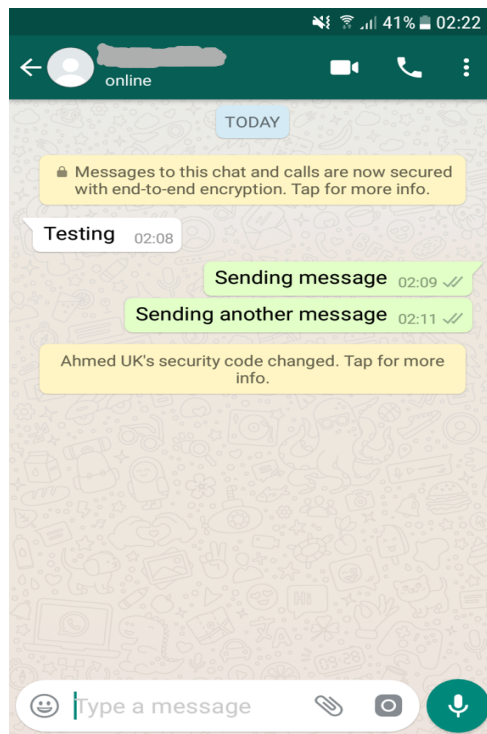
WhatsApp has around a billion users [1] and research has shown that users tend to stop adopting messaging applications with the slightest decrease in usability [10]. The usability Non-Blocking property of WhatsApp is therefore important. The attackers that can perform this attack can be WhatsApp owners or ordinary attackers. It is very risky for WhatsApp owners to perform these attacks since the attack can be detected if the user verifies the security code. WhatsApp will not risk that [4].



**Table 2: Comparison of Conversation Security in WhatsApp and Signal**

Conversation Security		WhatsApp	Signal
Security Properties	Confidentiality	✓	✓
	Integrity	✓	✓
	Authentication	✓	✓
	Participant Consistency	✓	✓
	Destination Validation	✓	✓
	Forward Secrecy	✓	✓
	Backward Secrecy	✓	✓
	Anonymity Preserving	✗	✗
	Global Transcript	✗	✗
	Message Repudiation	✓	✓
	Participant Repudiation	✓	✓
	Open source	✗	✓
	Message Unlinkability	✓	✓
Usability Properties	Out of Order Resilient	●	●
	Asynchronous	✓	✓
	Multi-device Support	✓	✓
	No Additional Service	●	●

✓ = Fully Satisfied  
● = Partially Satisfied  
✗ = Not satisfied



**Figure 2: The way WhatsApp and Signal handle message retransmission when user's security code changes**





**Figure 3: The new title of WhatsApp Polemic**

Additionally, other attackers will need to have a lot of information about the user before they can perform this attack. They have to know when Alice will send the message and they have to make sure that the Bob is offline or they have the ability to stop the message from being delivered. Additionally, this attack cannot be used for surveillance. It is, therefore, safe to say that the vulnerability is a reasonable tradeoff.

## 6. CONCLUSION

In this work, we presented the process through which secure messaging applications handle end-to-end encryption. That is through tackling three major problem areas: Trust Establishment, Conversation Security and Transport Privacy. At each goal is to achieve certain desirable properties in security and usability. Constraints from technology and conflicting requirements implies that tradeoffs have to be made.

We studied how two messaging applications, WhatsApp and Signal, implemented a secure messaging protocol called Signal. The similarities and differences in how these two applications approached achieving the desired security protocols and tradeoff made were discussed. Some implementation decision results in vulnerability, but since exploitation of the vulnerability is very difficult due very low probability of all the desired conditions happening at the same time for the attack to succeed, the decision is acceptable.

Future work will focus on analysing the Signal protocol that both WhatsApp and Signal applications rely upon in implementing the secure messaging services. The focus would be on the strengths and weaknesses of the protocol and possible improvement.

## REFERENCES

- [1] About whatsapp. <https://www.whatsapp.com/about/>. Accessed: 2017-09-09.
- [2] Flawed reporting about whatsapp. <https://www.theguardian.com/technology/commentisfree/2017/jun/28/flawed-reporting-about-whatsapp>. Accessed: 2017-09-09.
- [3] Signal. <https://signal.org/#page-top>. Accessed: 2017-09-09.
- [4] There is no whatsapp backdoor. <https://whispersystems.org/blog/there-is-no-whatsapp-backdoor/>. Accessed: 2017-07-01.
- [5] Whatsapp backdoor allows snooping on encrypted messages. <https://www.theguardian.com/technology/2017/jan/13/whatsapp-backdoor-allows-snooping-on-encrypted-messages>. Accessed: 2017-07-01.
- [6] Whatsapp design feature means some encrypted messages could be read by third party. <https://www.theguardian.com/technology/2017/jan/13/whatsapp-design-feature-encrypted-messages>. Accessed: 2017-09-09.

- [7] Mihir Bellare, Kenneth G Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In *International Cryptology Conference*, pages 1–19. Springer, 2014.
- [8] Tobias Boelter. Whatsapp retransmission vulnerability. <https://tobi.rocks/2016/04/whatsapp-retransmission-vulnerability/>. Accessed: 2017-07-01.
- [9] Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila. A formal security analysis of the signal messaging protocol. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, pages 451–466. IEEE, 2017.
- [10] Pui-Lai To, Chechen Liao, Jerry C Chiang, Meng-Lin Shih, and Chun-Yuan Chang. An empirical investigation of the factors affecting the adoption of instant messaging in organizations. *Computer Standards & Interfaces*, 30(3):148–156, 2008.
- [11] Nik Unger, Sergej Dechand, Joseph Bonneau, Sascha Fahl, Henning Perl, Ian Goldberg, and Matthew Smith. Sok: secure messaging. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 232–249. IEEE, 2015.
- [12] WhatsApp. Whatsapp encryption overview: Technical white paper. <https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf>. Accessed: 201707-05.