Unfriendly Chat

Team Wu-Tang LAN

Leslie Zhou (team lead)
Khanh Nguyen
Warren Singh

April 23, 2022

Contents

1 Introduction

In 2013, Edward "Ed" Snowden, a 29 year old government contractor who was a former technical assistant for the CIA and current employee of defense contractor Booz Allen Hamilton came forward with startling revelations: the United States government was indiscriminately collecting and spying on the internet communications of a huge majority of the English speaking world [1]. While security concerns are part of network engineering, major service providers and technology companies did not typically prioritize security at the time. Due to the revelations, securing their services and communications became a top priority, as the scope and extent of the 'bulk collections' programs that were being run by the NSA shocked even industry insiders.

Within six months, prominent companies such as Facebook, Twitter, and Google began implementing upgrades to both internal and external systems [2], and many consider this new approach to be the reason for the quick and widespread adoption of stronger security and end-to-end encryption protocols [3].

But how do technology companies actually secure communications and services for their users? Users will be less likely to use services which do not offer security and privacy, and in general societies are thought to suffer when they cannot protect the privacy of their citizens.

One open source cryptography project [4] is an industry leading standard [5] for end-to-end encryption, developed in the wake of the Snowden revelations: the Signal Protocol. The Signal Protocol [6] is a non-federated cryptographic protocol which is most widely used to ensure end-to-end encryption for communication applications (i.e. text-based messaging and VoIP). Applications which currently implement the Signal Protocol include Google's Messages, Facebook Messenger, Whatsapp, and Skype [5], meaning the number of users whose messages are secured by the Signal Protocol potentially number in the billions. (this matches the scope of the problem, as there are billions of users of electronic technologies around the world)

Due to its widespread use, broad influences, intended effect, and open sourced approach, examining the protocol thoroughly is crucial in understanding how industry leaders secure both internal and external network communications, as well as providing a foundation for apprehending and developing further iterations and applications, since developers working on applications continue to use the Signal Protocol as foundation and inspiration for further encryption protocol development.[7][8][9][10] Through the course of this project by which we implement the protocol in a real-time chat application setting, we seek to gain an understanding of this industry standard technology, and transmit that to our colleagues for their benefit as well.

The remainder of this report will details the overview, methods, and results of the implementation of the Signal Protocol. A high level understanding of the protocol is available to the casual reader, while others may wish to examine the citations for further exploration.

2 Project Description

2.1 Overview

The actual implementation of the project involved the connecting of a user facing react web chat application with the actual implementation of the cryptographic protocols and algorithms which were hidden from the user.

A separate server instance was used to store the pre-keys for the initial part of the Signal protocol (which is elaborated on in detail in the Methods section immediately following).

Various cryptographic primitives are relied in the course of the implementation of the actual cryptographic protocols, specifically public/private key pairs for signing from elliptic curve 25519 functions, AES 256 bit encryption for cleartext/ciphertext conversion concerning the user messages, and HKDF for the key derivation (so-called 'ratcheting') functionality. (more details on these in the methods section which follows)

A high-level illustrative flowchart of the project overview may be seen on the next page.

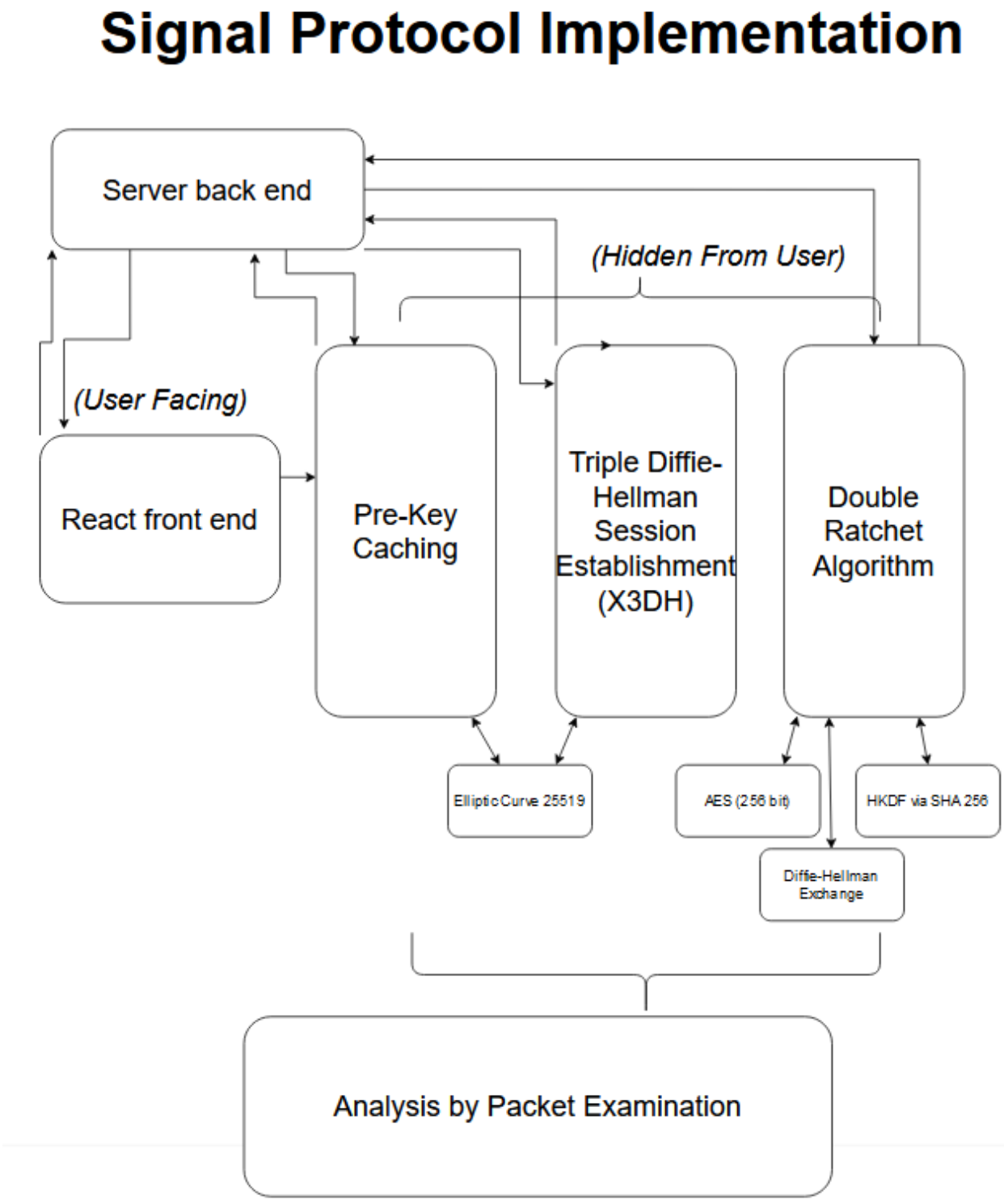

Figure 1: A flowchart illustrating an overview of the overall project design

2.2 Method(s)

2.2.1 React front end

2.2.2 Server back end

2.2.3 Pre-Key Caching

2.2.4 Triple Diffie-Hellman exchange

2.2.5 Double Ratchet Algorithm

2.2.6 Analysis by Packet Examination

3 Demo/Evaluation

3.1 Experimental Setup

3.2 Results

4 Conclusion and Future Work

5 References

References

[1] G. Greenwald, E. MacAskill, and L. Poitras. *Edward Snowden: the whistleblower behind the NSA surveillance revelations*. June 2013 [Online]. Available. URL: https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance.

[2] C. C. Miller. *Angry Over U.S. Surveillance, Tech Giants Bolster Defenses*. Nov. 2013 [Online]. Available. URL: https://web.archive.org/web/20131106015230/https://www.nytimes.com/2013/11/01/technology/angry-over-us-surveillance-tech-giants-bolster-defenses.htm.

[3] A. Cuthbertson. *Snowden "Sped Up Encryption" by Seven Years*. June 2016 [Online]. Available. URL: https://www.newsweek.com/snowden-sped-encryption-seven-years-452688.

[4] C. Garling. *Twitter Open Sources Its Android Moxie*. Dec. 2011 [Online]. Available. URL: https://web.archive.org/web/20111222010355/http://www.wired.com/wiredenterprise/2011/12/twitter-open-sources-its-android-moxie/.

[5] J. Lund. *Signal partners with Microsoft to bring end-to-end encryption to Skype*. Jan. 2018 [Online]. Available. URL: https://web.archive.org/web/20200202152037/https://signal.org/blog/skype-partnership/.

[6] M. Marlinspike. *Signal Technical Information, Specifications, and Documentation*. [Online]. Available. URL: https://signal.org/docs/.

[7] O. Eyal. *Canada, Germany and Australia are getting e2e encryption*. May 2016 [Online]. Available. URL: https://web.archive.org/web/20161005083000/http://www.viber.com/en/blog/2016-05-03/canada-germany-and-australia-are-getting-e2e-encryption.

[8] (unavailable). *Viber Encryption Overview*. [Online]. Available. URL: https://web.archive.org/web/20160711035838/http://www.viber.com/en/security-overview.

[9] J. Mayfield. *Forsta developer AMA (interview)*. Apr. 2018 [Online]. Available. URL: https://web.archive.org/web/20180502045526/https://www.reddit.com/r/crypto/comments/8b1m6n/forsta_signal_based_messaging_platform_for/.

[10] J. Mayfield. *Forsta codebase, (Github repository)*. July 2019 [Online]. Available. URL: https://web.archive.org/web/20180613054634/https://github.com/ForstaLabs/libsignal-node.