CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

The Signal Protocol for non-Cryptographers

An Explanation of the Signal Protocol and its Security Properties

Master's thesis in Computer Science - algorithms, languages and logic

Lamiya Yagublu

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018

The Signal Protocol for non-Cryptographers

An Explanation of the Signal Protocol and
its Security Properties

Lamiya Yagublu

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

The Signal Protocol for non-Cryptographers
An Explanation of the Signal Protocol and its Security Properties
Lamiya Yagublu

Supervisor: Gerardo Schneider, Department of Computer Science and Engineering
Advisor: Elena Pagnin, Carlo Brunetta, Chalmers University of Technology
Examiner: Andrei Sabelfeld, Department of Computer Science and Engineering

Master's Thesis 2018
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2018

Investigating the Security of the Signal Protocol
An Explanation of the Signal Protocol and the its Security Properties
Lamiya Yagublu
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

People tend to socialize and today many people use messaging applications to communicate. While people communicate, they share personal information between each other and they do not want others to observe or access their information and use it against them. Therefore, it is important to keep this information private. The Signal protocol is a communication protocol used to provide security guarantees and keep the users' information private while they communicate. Since many messaging applications, including WhatsApp and Facebook messenger, uses the Signal protocol and a lot of people use these applications, any flaw in the Signal protocol might affect a large number of users' private communication.

This master thesis aims to study the Signal protocol and explains, in an easy way, its functionality and security properties. The thesis contains a detailed explanation of the core parts of the Signal protocol and its security. A literature review was conducted to investigate how the Signal protocol works and what kind of security properties it has. Due to complexity of the problem in the few academic papers, the Signal protocol is defined and explained in an easier way. The thesis focuses mainly on the academic paper titled *"A Formal Security Analysis of the Signal Messaging Protocol"* by K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila.

As a technical contribution of this thesis, the existing notation from the above mentioned paper is revisited and simplified. Thereafter, the new and simplified notation is used to explain how the Signal protocol works in an easy way. The explanation with the simplified notations helps non-technical people to understand the protocol better. Then the security properties of the Signal protocol are investigated and explained.

Keywords: **Signal, instant messaging, key distribution system, double ratchet, forward Secrecy, post-compromised security, end-to-end encryption, authenticated encryption with associated data**

Acknowledgements

This master thesis project was conducted during the period from February 2018 to November 2018.

First of all, I would like to thank my father Ziyadkhan Yagublu, my mother Suraya Yagublu and my husband Asif Shukurov who have motivated and supported me during my studies at Chalmers. I also would like to thank the Adlerbert Foundations which provided me with the Adlerbert Study Scholarship and gave me an opportunity to be a part of Chalmers University of Technology.

Secondly, I would like to thank PhD. student Elena Pagnin for suggesting and encouraging me to choose this topic and helping me with administrative part of the thesis work.

I would also like to thank Prof. Gerardo Schneider, PhD. students Elena Pagnin and Carlo Brunetta for supervising and guiding me in writing this thesis and my examiner Prof. Andrei Sabelfeld for helping me. I am specially grateful to PhD. students Elena Pagnin and Carlo Brunetta, for taking time to answer my questions and priceless help. Without their feedback, I would not reach this far.

Finally, I would like to thank my friend Hikmat Hajiyev who have read my thesis report and gave his feedback on how to make it better.

Lamiya Yagublu, Gothenburg, November 2018

Contents

List of Figures

List of Tables

1

Introduction

Because of people's tendency to socialize, they like communicating. Nowadays, we have a lot of social networks that put people in contact. Even though the social networks help people to communicate with each other, they also bring some problems. When we register ourselves in a social network, we agree to make our data reachable to other people around us.

In recent time, Snowden revelations and Facebook Cambridge Analytica scandal showed that internet users' information might be monitored or accessed by other people to whom the users' do not give consent [60, 15]. the collected data can be used for different purposes like interfering with the presidential elections. This is the case of Facebook-Analytica scandal in which a company called Cambridge Analytica [8], accessed and used Facebook user's personal data in order to subliminally modify the users' political beliefs and influence the political election. People do not want to get surveilled in their personal life. Therefore, keeping personal information is important. Everybody who is concerned about it tries to find ways to solve this problem.

One way is introducing **secure communication protocols**. By using cryptographic primitives, secure communication protocols confirm the authenticity of people who are involved in the communication and guarantee confidentiality and integrity of information transmitted between these people [32]. The Signal protocol is one of them. The Signal protocol is a communication protocol which ensures that personal data of users is not interfered. It was introduced by software organization Open Whisper Systems in 2014 and now deployed by WhatsApp [67] and Facebook Messenger [36].

Since Facebook has 2.07 billion monthly active users [42], any flaw in the Signal protocol might affect a large number of users' private communication. Therefore, it is of at most importance to understand how the Signal protocol works and investigate the security of the Signal protocol.

The Signal protocol has been studied from academic point of view and there are several papers that explains how the Signal protocol works and what kind of security properties it has. Although there are academic papers about the Signal protocol and its security, none of them are explained for non-technical people. These papers use complex notation and figures to describe the Signal protocol. Papers like [11, 18, 9, 26, 40] explain the Signal protocol with different notation. Some of them explain the Signal protocol based on information either from the original documentation [55] or from academic paper [11]. Cohon-Gorden et al's paper [11] has the more detailed explanation of the Signal protocol but it has difficult language and complex

notation that do not allow ordinary readers to understand the content easily. This lack makes the topic of the master thesis more appealing.

1.1 Privacy

Attempts on protecting one's own personal information from others' have become one of the biggest concerns and a part of modern life. Almost every day people come across news about threats on internet user's personal information. We are exposed to many news about privacy, but *can one define or at least explain in an easy way what privacy is?*

According to the Oxford Dictionary, the meaning of privacy is:

1. A state in which one is not observed or disturbed by other people.
2. The state of being free from public attention.

which can be seen as a pretty informal definition. Stallings and Brown define it as "**Privacy** assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed." [64].

For example, Google collects users' information for different reasons such as communicating, protecting, and providing personalized content to users. Google privacy policy explains what type of user information are collected and used by Google and explains how users can manage, review, update, export, and delete their information. Users can control their personal information, share information with others and information generated while using any of Google services [19].

Why is privacy important? The growth of technology makes everybody to be concerned about the privacy. But why one need to think about it? Why is it important? Some people think that losing sensible information or personal data is just annoying and irritating. However, the violation of the privacy matters a lot more than that. The privacy limits everybody and everything around to get the personal data that belongs us. This means one is protected from violation of the personal data and impact that can affect the owner of the information in a bad way. It is a fact that the more someone knows about us, the more power they can have over us. Most obvious example could be friends. They usually know more about us than everyone else, and they can use this data against us. If one knows about us they can control us psychologically, affect our reputation and influence our decisions. These all directly mean that our privacy may have power to control our life, which is awkward and unacceptable.

For example, in October, 2017 an American restaurant chain **Pizza Hut** informed some of their customers that the hackers have accessed their sensitive data. This data includes their names, addresses, and credit card information, such as bank account numbers, expiration dates of the credit cards and Card Verification Value numbers. According to some customers of Pizza Hut their credit card was used by hackers since Pizza Hut informed their customers too late [44].

Web privacy. Over the years, the number of the internet users increases as it is

possible to see in Table 1.1. Nowadays many people cannot imagine their life without internet [7]. People communicate, do online shopping, and gather information over the internet. This means millions of people can access, steal or damage our private information. Therefore, every individual should think how to protect their data in this huge web of information. For instance, if our credit card information is stolen while we do online money transaction, then the money in our bank account might be stolen as well.

World Regions	Population (2018 Est.)	Internet Users 31 DEC 2017	Growth 2000-2018 %	Internet Users %
Africa	1,287,914,329	453,329,534	9,941 %	10.9 %
Asia	4,207,588,157	2,023630,194	1,670 %	48.7 %
Europe	827,650,849	704,833,752	570 %	17.0 %
Latin America	652,047,996	437,001,277	2,318 %	10.5 %
Middle East	254,438,981	164,037,259	4,893 %	3.9 %
North America	363,844,662	345,660,847	219 %	8.3 %
Australia	41,273,454	28,439,277	273 %	0.7 %
World Total	7,634,758,428	4,156,932,140	1,052 %	100.0 %

Table 1.1: World Internet Usage and Population Statistic December, 2017 [24].

Data privacy must be regulated as every data holder want to be assured that their data is protected.
Since we want to protect our data, there should be some rules and regulations for data privacy and they should be applied to all kind of users: private individuals, companies, organizations, government and etc. Privacy law contains the laws that controls the personal data that can be accessible to a third party. It is essential to have some laws and regulations to protect data that occurs on the web as well. Privacy law also contains the rules regarding the internet privacy where these rules are applied to the information exchange that happens through internet.

The European General Data Protection Regulation (or GDPR) [43] is a set of rules that explains when and how companies or organizations are allowed to collect or use the personal data that belong to people. It was designed to protect the private information across Europe. This law makes sure that firms ask for consent of users to collect their data, inform them about how they keep this data secure, notify them in case of any data breaches, allow them to move their data to another organization, or delete the data upon the request of users. Thus, people started to have more control over their data and they know what data are gathered by firms and why they gather this data [68].

Examples. History has a lot of examples when the privacy violation has become a great scandal. One can say it is awkward. But experience, even if it is negative, teaches us. These scandals became a main reason for every individual to think about their privacy, for IT companies to improve their product from the security point of view and government to improve data protection systems, to apply new rules and regulations related to privacy.

Snowden Revelation. One of the biggest security scandals happened in the recent years is known as "Snowden revelations" [61]. In 2013 NSA contractor Edward Snowden revealed that National Security Agency (NSA) and British Intelligence and Security organization (GCHQ) illegally collected, stored and analyzed vast amount of data from citizens, officials and big companies across the globe. They tapped phone calls, internet use, and short messages of people gathering sensitive information such as financial data, contacts, and location. Operations also included hacking networks related to Chinese officials and businesses, spying on EU offices, monitoring e-mails and phone calls of Latin America leaders, and conducting surveillance operations on embassies of foreign countries in USA [13]. NSA also illegally reached and collected emails, photo and video content from giant technology companies such as Facebook, Google, and Yahoo [17].

As a reaction to this scandal, the US Senate adopted a new law called *"USA Freedom Act"* which banned any party from gathering bulk collection of phone records [41]. Similarly, in UK, GCHQ´s surveillance operations was ruled as illegal by UK Investigatory Powers Tribunal which breached European Commission articles about human rights [17]. This scandal also increased public awareness about private data, its collection and usage by other parties. As a result, people changed their usage behaviour to be able to protect their privacy better. For example, internet users started to increasingly use email encryption technologies and anonymous browsing tools [53].

Big technology companies including Facebook, Google, and Yahoo stated that they did not allow governments to access their data centres. They changed their privacy policies and started to inform users more regarding data demands by government unless it is prohibited by court order [4]. These companies increased encryption across their service platforms as a quick countermeasure to government surveillance operations. For example, Google and Yahoo used end-to-end encryption for their mail services which means that the message remains encrypted all the way from the starting server to the destination server. This encryption method even prevents the companies themselves to access users' inboxes. Facebook also took encryption countermeasures which allowed users to visit Facebook privately and securely using Tor browser [53].

Facebook Analitica Breach. Another major data scandal happened in early 2018 [16]. Through a third-party application, implemented as a Facebook's psychological test, a researcher collected private data of 87 million Facebook users. Due to a loophole in the system this application could access the private data of users taking the test and of people in their friend list. Then the researcher sold this data to a consulting company Cambridge Analytica which targeted US presidential elections and Brexit referendum. This data scandal caused a lot of concerns about the privacy of Facebook users and how their data are handled [58].

Directly after this data scandal, General Data Protection Regulation (GDPR) went into effect in EU. Accordingly, firms operating in EU countries updated their privacy policies and notified their customers about these updates. The US government is lagging behind regarding adoption of comprehensive data protection law similar to GDPR. US Federal Trade Commission (FTC) just investigate to see if

Facebook violated an agreement between them over protecting privacy of Facebook users. Even though the US and other countries lagging behind to adopt a similar law to GDPR, many firms plan to apply GDPR requirements globally [68].

The US government is rather working on more specific bill like *"The Honest Ads Act"*, which will regulate online political and issue advertisements. This law force social network companies to reveal the location of people running political and issue advertisements, their target audience, and their financial source. This increases awareness among users and help them to avoid from exposure of these campaigns, if they want [65]. Facebook has already proactively created similar public archive of political advertisements [30]. Facebook also started to use more advanced artificial intelligence tools to detect fake accounts and prevent misinformation [46]. Besides, Facebook took further actions to protect the privacy of its users. Now they work on a *"clear history"* tool which will give users more visibility and control over what applications accessed their data and delete this information from there account [31].

1.2 Digital Communication

The twenty first century is a time when humanity is surrounded by technology. Almost every sector of the industry, our everyday life depend on technology. The use of digital technologies helps us to communicate. Digital communication is part of our life. Digital communication is a mode of communication that converts the information into digital format, *i.e.,* computer readable and transmits the information to other parties.

Why digital communication is popular and important nowadays? Why is there so much need for it? The reasons are:

- It is the quickest type of communication. For instance, one wants to share files, it will take long time if that file will be sent by regular mail. Of course, there is an option of paying some money to decrease the time, but in that case the financial aspect of it makes this option expensive. However, sending files will take several minutes, even seconds and additionally, it will be less resource demanding, more environmentally friendly, and financially viable if the digital communication methods are used;
- Digital communication allows people across the world to communicate virtually face to face. For example, there are a lot of people who travels to other countries to study or work. They can easily call their relatives or loved ones, hear their voice and see them on the screen of the device;
- Every individual can be a part of the influence that helps to make better world and future. Digital communication provides us with environment and platforms where we can share our ideas, be a part of the social media, and collaborate with other people even if we are from different cultures.

Digital communication is ensured by digital communication protocol, and devices. The information about one of the communication protocols will be discussed later in Chapter 3. Now lets have look at digital communication methods and devices.

1.3 The Digital Communication Methods and Devices

In this part of the thesis, the methods and devices that ensure the digital communication are discussed with examples and their advantages and disadvantages.

The first method is communicate using a **forum**. A forum is an online environment where people join in order to share their ideas and views about a particular topic. In forums, the messages are in the form of posts. For example, **Stack Overflow** is one of the most popular forums among the people who have any connection to computer science, engineering and information technologies [51]. Another popular forum is **Digital Photography Review** which is used to discuss different photography topics and display product reviews [12]. The advantage of forums is that it is interesting for users since the conversation in the forum is always about a specific topic. Eventually, users can find a topic or topics they are interested in and read all information about that topic. The disadvantage of forums is that any reply on a forum is not done in real-time. It is easy to access the forums using computers, smartphones and tablets.

The second method is **weblogs**. The weblogs involve lists, text, or objects that are maintained, arranged and run by a single person. The other characteristic of the weblogs is that they usually have chronological order where the last post appears first. For example, **TechCrunch** [54] is a weblog about the industry of technology with news related to technology. It also contains analysis of new technology trends, products and businesses. The advantage of the weblogs is that the owner of a weblog can do whatever he/she wants such as manage, post, delete or change the information. Containing subjective views can be a major disadvantage for a weblog since the information is provided only by the owner. Weblogs can also be accessed on devices like computers, smartphones and tablets.

The third method is **wikis**. The Wikis allow people around the world to collaborate, add, delete, or change the information. One can say that wikis are the same as weblogs. But it is not true since the main difference is that weblog has an owner who creates it and manages all information while wikis do not have any specific owner. For example, **WikiTravel** is a wiki which has all the information related to traveling [69]. The advantage of wikis is that one can get a lot of useful information. The disadvantage of wikis is that the information on this web applications can be modified by any person which makes wikis less reliable. If one has a computer, smartphone or tablet, he/she can access wikis easily.

The fourth method is **Electronic mail** (Email). Email is the way of exchanging digital messages. One can deliver, accept, store, delete and forward messages using email. For instance, **Google Mail**. Email is an effective and practical way of transferring the information. The disadvantage of email is that there can be delays in replies since it is not done in real-time. Another disadvantage is the slightly complicated structure where multiple components such as email clients of sender and receiver (and respective mail servers) need to be configured and running.It is also accessible on computer, smartphones and tablets.

The fifth method is **Instant Messaging** (IM). IM is a chat that provides mes-

sage or data transmission in real-time. For example, WhatsApp. The advantage of IM is that people communicates in real-time. The disadvantage is this communication in real-time limits the face-to-face communication which both sides can give the information that they want other party to know. IMs are usually accessed by smartphones only but mostly using smartphones, but some of them have versions developed for computers.

Protocols. To control the communication a set of rules are used. This set is called a protocol. A protocol can be implemented by using hardware, software or both of them. Since the thesis focuses on Signal protocol, we are interested in protocols that are implemented in software. The protocols has difference according to what and how they manage and control. For example, internet protocols (TCP/IP) and cryptographic protocol (Signal protocol). An internet protocol is a group of rules that applied on communication across the network, but cryptographic protocols are the set of rules that ensures secure communication. The Signal protocol is a cryptographic protocol. The cryptographic protocol is defined in the book titled "Handbook of Applied Cryptography" [1] as "A **cryptographic protocol** is a distributed algorithm defined by a sequence of steps precisely specifying the actions required by two or more entities to achieve a specific security objective.".

Instant Messaging. According to [27, 70, 62, 52, 25], the popularity of instant messaging (IM) is increasing day by day since it is an easy way to communicate and keep in touch with other people.The most popular IM apps like WhatsApp [39], Facebook Messenger (see Table 1.2) have several benefits for the users as stated in [70, 52]:

- Users can easily send requests and get responses;
- Users are provided with features like presence and event notification that is used for tracking of the availability of other people;
- Users can make voice or video chats, send files;
- Users can get messages in real-time. In that case, all of the communicating parties should be online.

IM apps	Monthly Active Users in Millions
WhatsApp	1500
Facebook Messenger	1300
Skype	300
Viber	260
SnapChat	255
Telegram	200

Table 1.2: Most popular mobile messaging apps worldwide as of july 2018, based on number of monthly active users (in millions).

In IM protocol, users have unique names, and list of friends. If one or more users are online, IM system notifies about that to their friends and they can communicate

while they are online [70]. Considering its features mentioned above, IM is an effective and efficient real-time, text-based private communication which transmits texts between two or more users [25, 5].

IM Apps. When one buys a smartphone, he/she usually downloads an IM app which helps to chat and make a voice call. It does not matter which operating system runs on that smartphone, one can still access some IM app. Lets have a look to some popular IM apps that are used all over the world (see Table 1.3).

App Name	App Information
Messenger	It is also called as Facebook Messenger. This app provides Facebook users with real-time communication. People can communicate via text, voice and video calls. It also has features like sharing stories, chatting in a group [35].
WhatsApp	It is owned by Facebook. Instant messaging, voice and video calls, sharing photos, videos, documents and location are available on WhatsApp platform. Group chats is one of the features that WhatsApp users are provided with [66].
Skype	Skype is also used to communicate using messaging, voice and video call. People usually use Skype for business purposes. For example, most of the international interviews are made via Skype or there are people that offer online classes via Skype [49].

Table 1.3: Information about Instant Messaging Apps.

We can divide IM into two categories according to the design of the conversation system: **synchronous** and **asynchronous**. In synchronous communication, the participants should be online, but in asynchronous communication, it is not a requirement [20].

The Signal protocol is an example of the asynchronous protocols.

1.4 History of the Signal Protocol

In 2010, the startup company called Open Whisper Systems (OWS) developed and published **TextSecure protocol** [3]. There were three versions of the TextSecure protocol:

1. **TextSecure v1** was an OTR (Off-the-Record) based protocol which was the combination of AES symmetric-key algorithm, the Diffie–Hellman key exchange, and the SHA-1 hash function [11].
2. **TextSecure v2** used the algorithm called Axolotl Ratchet and was introduced in February 2014 [63, 11].
3. **TextSecure v3** had some changes and improvements to TextSecure v2. This version was introduced in October 2014 [52, 11].

In March 2016, The TextSecure protocol v3 was named Signal [11, 47]. In October 2016, there was some cryptographic changes to the Signal Protocol [11].

Nowadays mobile messaging applications such as Signal, WhatsApp, Messenger use the Signal protocol.

1.5 Aim of the Thesis

This master thesis project aims to study the Signal protocol in detail and explain its functionality and properties in a way accessible to people without a cryptography background.

The aim of the master thesis project can be broken into the following research questions:

RQ1: How does the Signal protocol work?

RQ2: Is the Signal protocol a secure protocol for in social network applications?

RQ3: What kind of cryptographic primitives and techniques does the Signal protocol use to achieve privacy and security?

This is an academic contribution, since there is only few paper published in a peer-reviewed venue [11] and further formal study and/or presenting has to be made.

1.6 Content

The thesis is divided into the several chapters. Chapter 2 contains all information about the background that is needed to follow the rest of the chapters easily. The Signal protocol is explained in Chapter 3. In Chapter 4, some security notions are discussed and explained. Chapter 5 will conclude the report.

2
Background

This chapter contains all the minimal mathematical and cryptographical background needed to understand the building blocks and design principle of the Signal protocol.

Section 2.1 contains a quick summary of Group Theory. Section 2.2 contains the fundamental information used to define Diffie-Hellman (DH). Section 2.3 contains a description of the Discrete Logarithm problem and the DH assumptions which will be helpful to understand some security properties in Chapter 5. Section 2.4 contains the explanation of one-way functions, hash functions and Message Authentication Code (MAC) which will make it easy to understand Hash-based Message Authentication Code (HMAC). Section 2.5 contains a description of Authenticated Encryption with Associated Data (AEAD) which used to exchange messages. Section 2.6 presents Key Derivation Functions (KDF) and key derivation chain that are used in the core part of the Signal protocol: *i.e.,* Ratchet.

The content of this chapter gives the reader a high-level concept on how the above mentioned cryptographic primitives work. The focus of the master thesis is to explain the DH key exchange protocol, HMAC, AEAD and KDF only up to the level needed to understand their use in the Signal protocol. Therefore, we will discuss and explain the information that will be helpful to understand the Signal protocol without going deep into mathematical notations, calculations and implementation details so that the intuition behind each of the above mentioned notions is enough for someone to understand how the Signal protocol works.

Before discussing the details of the background topics, we start with some terminology and notions that appear during this chapter.

The CIA Triad. We refer to Confidentiality, Integrity and Authentication as CIA which is security design principles. These objectives define the CIA Triad. *Confidentiality* is the concept of keeping the information secret and protecting it from unauthorized parties. *Integrity* is the process of preventing modification of the data by unauthorized parties. *Authenticity* is a prove and confirmation of being trusted and verified; being confident that a transmission, a message, or message originator is valid [64].

Alice and Bob. The Signal protocol is designed for secure communication. Therefore, in what follows we use two parties that communicate with each other. We use two parties Alice and Bob, this approach lets us show the communication, the information exchange and the examples in an easier and more intuitive way. Sometimes

we use expressions like *"Alice and Bob communicate over a **public channel**"* which means they communicate over a not secure channel.

Negligible. We express the small probabilities of success of an adversary using **negligible** (*negl*) which refers to probabilities (Pr) smaller than any inverse polynomial in n (security parameter). More formally,

Definition 1. [28] A function f from the set of natural numbers to the non-negative real numbers is **negligible** if for every positive polynomial p there is an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.

PPT Algorithm. PPT abbreviation stands for *"probabilistic polynomial-time"* which refers to algorithms that run in polynomial-time (efficient) and use coin tosses, producing randomized outputs.

2.1 Group Theory

In this Section, some basic definitions are provided which enable us to formalize and explain the notions which are the backbone to cryptographic primitive like DH (see Section 2.2).

Definition 2. [28] A group is a set $\mathbb{G}$, along with a binary operation $\circ$ for which the following conditions hold:

- **Closure:** For all $g, h \in \mathbb{G}$, the element of $g \circ h \in \mathbb{G}$;
- **Associativity:** For all $g_1, g_2, g_3 \in \mathbb{G}$, it holds that $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$;
- **Identity element:** There exists an element $e \in \mathbb{G}$ such that for all $g \in \mathbb{G}$, it holds that $e \circ g = g = g \circ e$;
- **Inverse element:** For all $g \in \mathbb{G}$, there exists an element $h \in \mathbb{G}$ such that $g \circ h = e = h \circ g$.

It is more common in cryptography to use either the additive notation, with "+" as operation, or multiplicative operation, "$\cdot$" as operation, instead of a generic binary operation $\circ$. In the following definitions we adopt the multiplication notation. We also use sign "|" for divisibility, *i.e.,* $a|b$ means *"a divides b"*.

Definition 3. [28] *The order of a element* a in a group $\mathbb{G}$ is the smallest positive integer n such that $a^n = e$, where e is the identity element of $\mathbb{G}$.

Theorem 1 (Lagrange Theorem)**.** [28] For every element a of a group $\mathbb{G}$, it holds that the order of a divides the order of $\mathbb{G}$, *i.e.,*

$$\mathbf{ord}(a) \mid \mathbf{ord}(\mathbb{G}).$$

Definition 4. [28] A non-empty subset H of a group $\mathbb{G}$ is said to be a **subgroup** of $\mathbb{G}$, if H is itself a group under the same operation of $\mathbb{G}$, *i.e.*, close under the multiplication, it contains e and the inverses for all its elements.

Definition 5. [28] Let $\mathbb{G}$ be a group and $g \in \mathbb{G}$. Define the subgroup generated by g as

$$< g >=< g^i \mid i \in \mathbb{N} > .$$

Definition 6. [28] An element g of a group $\mathbb{G}$ is called a **generator** of $\mathbb{G}$ if $< g >= \mathbb{G}$.

Definition 7. [28] A group $\mathbb{G}$ is **cyclic** if it has a generator, i.e., if there exists an element $g \in \mathbb{G}$ such that $< g >= \mathbb{G}$. Moreover, g is a generator of $\mathbb{G}$ if and only if $| < g > | = |g| = |\mathbb{G}|$.

Lemma 1. [28] Every element h of a cyclic group $\mathbb{G}$ can be written as a power of g, *i.e.*,

$$\text{for all } h \text{ in } \mathbb{G}, \text{ exists } k \text{ in } \mathbb{N}, \text{ so that } h = g^k.$$

Lemma 2. [28] Let $\mathbb{G}$ be a group with identity element e, and let $a \in \mathbb{G}$, then

$$a^{|\mathbb{G}|} = e.$$

Definition 8. [28] The **greatest common divisor** of $a, b \in \mathbb{Z}$ is denoted as $\gcd(a, b)$ and it is defined as the largest integer c such that $c|a$ and $c|b$.

Let us define the set of all integers in the set $\{1, ..., q-1\}$that are relatively prime to q as $\mathbb{Z}_q^* \stackrel{def}{=} \{b \in \{1, ..., q-1\} | \gcd(b, q) = 1\}$. We also define a set $\mathbb{Z}_q = \{0, ..., q-1\}$ with q elements.

2.2 The Diffie-Hellman Key Exchange Protocol

The Diffie-Hellman key exchange protocol (DH) is a protocol that is used to establish a secure communication channel between two parties. This means that two parties who do not know each other and have never agreed on any secret, can establish a shared secret over a public and insecure channel. The DH key exchange does not provide any authentication. However, many authentication protocols are based on it [28, 29].

Let $\mathbb{G}$ be a cyclic multiplicative group of order q with generator g. Let us consider a situation in which Alice and Bob want to securely communicate over a public channel. In order to establish a secure communication, they need to securely agree on some common shared secret and it is illustrated in Figure 2.1. The Diffie-Hellman key exchange protocol has the following structure [28]:

- Alice chooses a random value $x \leftarrow \mathbb{Z}_q$ and computes $h_A = g^x$. She sends h_A to Bob;
- Bob chooses a random value $y \leftarrow \mathbb{Z}_q$ and computes $h_B = g^y$. He sends h_B to Alice;
- When Alice receives h_B from Bob, she computes the shared secret $K_A = (h_B)^x = g^{yx}$;

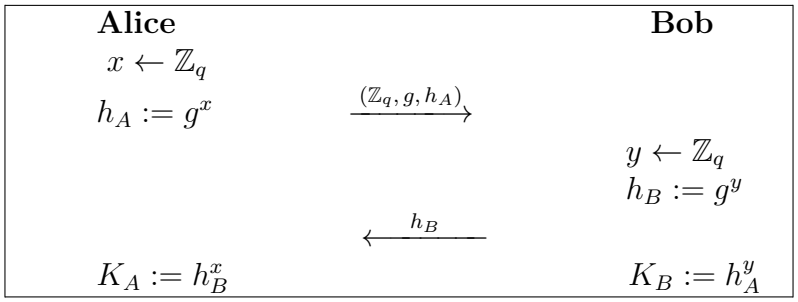

Figure 2.1: Diffie-Hellman Key Exchange Protocol between Alice and Bob.

- When Bob receives h_A, from Alice he computes the shared secret $K_B = (h_A)^y = g^{xy}$;
- At the end, Alice and Bob share a common secret namely the value $K_A = K_B \in \mathbb{Z}_q$ that can be used as a secret key for communication between them.

The Signal protocol uses the **Triple Diffie-Hellman (X3DH)** key exchange protocol to achieve some security properties. In order to have a shared secret, Alice and Bob obtain the shared secret by computations with their public keys. While Alice and Bob use X3DH they are also mutually authenticated. X3DH is designed in such a way that even if Alice is offline Bob can still compute a shared secret based on the public information that Alice has published to a server [38].

2.3 The Discrete-Logarithm Assumptions

Since the Signal protocol uses the triple Diffie-Hellman key exchange to ensure secure communication, this security should base on some security properties of Diffie-Hellman. Modern cryptography bases security on the assumption that a given problem is hard to solve in efficient way using computers. Such problems are referred to **computational hard problems** [28]. This section describes the most common computational hard problems on cyclic groups. Our main focus will be on two specific problems: the *discrete-logarithm* (Dlog) problem and the *Diffie-Hellman* problem.

Dlog Problem [29, 28]. Let $\mathbb{G}$ be a multiplicative group with generator g. The Dlog problem can be formulated as follows: for $h \in \mathbb{G}$, find $x \in \mathbb{Z}$ such that $g^x = h$. The assumption about this problem states that it is hard to solve Dlog problem for some cyclic groups.

The Diffie-Hellman Problem [28, 29]. The Diffie-Hellman problems are not equivalent, but are closely related to the discrete-logarithm problem. Let *log* be the **logarithm** as the inverse function of exponentiation, *e.g.,* if $log_g h_1 = x$ then $h_1 = g^x$. We divide the diffie-hellman problem into two :

- The computational Diffie-Hellman problem (CDH): Consider a cyclic group $\mathbb{G}$ with a generator g and $h_1, h_2 \in \mathbb{G}$. We define $\mathtt{DH}_g(h_1, h_2) \stackrel{def}{=} g^{log_g h_1 \cdot log_g h_2}$. If $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$, then $\mathtt{DH}_g(h_1, h_2) = g^{x_1 \cdot x_2} = h_2^{x_1} = h_2^{x_1}$. We want

to calculate $\mathtt{DH}_g(h_1, h_2)$ for uniformly chosen h_1, h_2 from $\mathbb{G}$. This problem is called CDH problem.

- The decisional Diffie-Hellman (DDH) problem: Given h_1, h_2 and h' that are chosen uniformly at random. We want to distinguish $h' = \mathtt{DH}_g(h_1, h_2)$ or decide h' was chosen uniformly from $\mathbb{G}$.

2.4 Hash-based Message Authentication Code

This section provides information about *one-way and hash functions, MAC* which will be helpful to understand the main concept of Section 2.4, HMAC.

Message Authentication Code. Message authentication code is crytographic primitive that provides data integrity and guarantees the identity of a sender. Assume Alice and Bob want to communicate using a MAC. First, they generate a key and share it. If Alice wants to send a message to Bob, she calculates an authenticator called **tag** using the message and the shared secret key and sends the message and that tag to Bob. In turn, Bob can check if the authenticator on the message is valid or not using the shared key [28, 22].

Definition 9. [28] A Message Authentication Code MAC is a triple of efficient algorithms **MAC=(**$KeyGen$**,** MAC**,** $Verify$**)** defined as follows:

- $\mathsf{KeyGen}(n) \rightarrow k$: is key generation algorithm which given a security parameter n outputs a secret key k.
- $\mathsf{MAC}(k, m) \rightarrow t$: is a tag (MAC) generation algorithm that takes as input a key k and a message m it outputs a tag t.
- $\mathsf{Verify}(k, m, t) \rightarrow \{1, 0\}$: is a deterministic verification algorithm that, given a key k and a tag t, it outputs 1 ("yes") if the tag verifies or 0 ("no") if it does not verify.

Correctness Property [28]: For all k, m and $t = \mathsf{MAC}(k, m)$, it holds that $\mathsf{Verify}(k, m, t) = 1$.

This means if a tag t computed based on a message m and a key k, then verification of the tag t based on the message m and key k outputs 1 which means it is a valid tag.

One-Way Functions. These functions are a cryptographic primitive that enables input to be indistinguishable even if output is known by other parties. In other words, although computing the output for a specific input is easy, it is difficult to retrieve the original input if the output is given [28].

Hash Functions. A crytographic hash function is one of the important primitives that is used to have secure authentication of data. Cryptographic hash functions contain a mathematical algorithm that takes some data as input and output a fixed size digest. For any input message, it is easy and fast to compute the digest using the hash functions. Hash functions are deterministic, it means they have the same

output digest for the same input message. However, computing the input message for the known digest is infeasible [28, 6].

HMAC. The Signal protocol uses HMAC to provide the integrity of the data. HMAC is a MAC mechanism that is based on cryptographic hash functions [22]. The security of HMAC depends on hash function it uses [23].

HMAC performs two hash computations. A secret key is used to derive two additional keys. The result of the first hash computation is derived from the first key and the message. Then this result together with the second key is used to derive the second hash computation. This kind of algorithm mitigates the length extension attacks (If an attacker knows the hash of the message $h(m_1)$ and the length of the message $l(m_1)$ he/she can calculate $h(m_1||m_2)$).

HMAC does not provide message encryption, instead a message is sent together with the HMAC hash. The party that receives the message with the HMAC hash and computes hash again herself/himself. if the received and computed hash are the same, then the received message is authentic [28, 22, 23].

The Signal protocol uses HMAC to verify the integrity of the messages and the authentication of messages. It uses HMAC-SHA256 and HKDF-SHA256 [11] where HKDF is an HMAC based on Key Derivation Function which is explained in Section 2.6 [21]. We will not discuss them since it is not a thesis topic. But it is good to know if anybody wants to learn more details about the Signal protocol.

2.5 AEAD

In this section, we will talk about Authenticated-Encryption with Associated Data (AEAD) which is a variant of Authentication Encryption (AE) - this cryptographic primitive ensures privacy and integrity of the message. An AEAD scheme provides the authentication of the additional information that is not encrypted but sent alongside with encrypted message [45]. In other words, AEAD ensures integrity and authentication of the encrypted message and authentication of the associated data [33]. In Signal protocol, AEAD is used to provide the authentication security properties [11]. Let us see the formal definition of the AEAD scheme. The sign $\perp$ that is used in the definition means false.

Definition 10. Let $\mathscr{M} = \{0,1\}^*$ be the message space of arbitrary long bit string, $\mathscr{C} = \{0,1\}^*$ be the ciphertext space of arbitrary long bit string and $\mathscr{K}^*$ be the key space of arbitrary long bit string. An authenticated encryption scheme with associated data, AEAD is a tuple $\Pi = (\mathscr{K}; \mathscr{E}; \mathscr{D})$ with an encryption algorithm $\mathscr{E}_K$ and a decryption algorithm $\mathscr{D}_K$. $K \in \mathscr{K}^*$ denotes the key, $H \in \mathscr{H}$ the associated data (or header), $N \in \mathscr{N}$ the nonce [1], $M \in \mathscr{M}$ the message, $T \in \mathscr{T}$ the authentication tag, and $C \in \mathscr{C}$ the ciphertext, where $K \subseteq \{0,1\}^k$, and $N \in \{0,1\}^n$, $\mathscr{T} \subseteq \{0,1\}^t$ denote the key, header, message, ciphertext, nonce and tag space, respectively, with $k > 1$ and $t > 1$. We write

[1] A nonce is an auxiliary input, representing a number used only once [14].

$$\mathscr{E} : \mathscr{K} \times \mathscr{H} \times \mathscr{N} \times \mathscr{M} \to \mathscr{C} \times \mathscr{T}$$
$$\mathscr{D} : \mathscr{K} \times \mathscr{H} \times \mathscr{N} \times \mathscr{C} \times \mathscr{T} \to \mathscr{M} \cup \{\perp\}$$

for the encryption function $\mathscr{E}$ and the decryption function $\mathscr{D}$ where $\perp$ means that *"something went wrong"*.

According to definition, (C, T) is a single string output C, is the output of encryption of message M where header H and nonce N is used under the key K. Some AE schemes provide the ciphertext and a tag separately, not in the form of single string. Decryption in this definition, returns either a message $M \in \mathscr{M}$, or $\perp$ which means false or *"the tag is not correct"* [14].

2.6 Key Derivation Function

In this section, we will introduce Key Derivation Function (KDF) since it is a part of the double ratchet algorithm (see Chapter 3) which is used by the Signal protocol [11]. Before talking about the KDF, we need to know what a Pseudo Random Function (PRF) and a Pseudo Random Generator (PRG) is [28].

Definition 11. Let l be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n and any input $s \in \{0,1\}^n$, the result $G(s)$ is a string of length $l(n)$. We say that G is a pseudorandom generator if the following conditions hold [28]:

1. **Expansion:** For every n it holds that $l(n) > n$.
2. **Pseudorandomness:** For any PPT algorithm D, there is a negligible function $negl$ such that

$$|Pr[D(G(s)) = 1] - Pr[D(r) = 1]| \leq negl(n),$$

where the first probability is taken over uniform choice of $s \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $r \in \{0,1\}^{l(n)}$ and the randomness of D.

Let G_0, G_1 be functions denoting the first and second halves of the output of G; *i.e.*, $G(k) = G_0(k)||G_1(k)$ where $|G_0(k)| = |G_1(k)| = |k|$. Now a PRF can be constructed [28]:

Let G be a pseudorandom generator with $l(n) = 2n$, and define G_0, G_1 as in the text. For $k \in \{0,1\}^n$, define the function $F_k : \{0,1\}^n \to \{0,1\}^n$ as: $F_k(x_1x_2...x_n) = G_{x_n}(...(G_{x_2}(G_{x_1}(k)))...)$.

Figure 2.2: A PRF construction.

A **Key derivation function** is one of the important components of the Signal protocol. Signal's double ratchet algorithm uses the KDFs in a chain. This chain

is used to update the keys while new messages sent and received. It is a function that takes random initial keying material and some input data and returns secret key using a PRF[37, 55]. The KDF is defined as follows [21]:

Definition 12. A **key derivation function** accepts as input four arguments: a value σ sampled from a source of keying material, a length value l, and two additional arguments, a salt value r defined over a set of possible salt values and a context variable c, both of which are optional, i.e., can be set to the null string or to a constant. The KDF output is a string of length l.

2.7 Digital Signatures

In this section, we will discuss digital signatures which ensure the integrity of the sent message or data and the authenticity of sender (they are the MAC, but in the public key domain).

Assume Alice wants to send a message to Bob and she has a public key pk and a secret key sk known just by her. When Alice sends a message, she uses sk to sign the message. When Bob receives the message, he uses the pk associated to the sk used to sign the message and verifies that the message has been sent by Alice.

Definition 13. [28] A (digital) signature scheme is a triple of PPT algorithms ($KeyGen$, $Sign$, $Verify$) defined as follows:

- $\mathsf{KeyGen}(\lambda) \rightarrow (pk, sk)$: is a key generation algorithm which given a security parameter λ outputs a public and a private key (pk, sk) pair.
- $\mathsf{Sign}(sk, m) \rightarrow \sigma$: is a signing algorithm that, given a secret key sk and a message m, it outputs a signature σ.
- $\mathsf{Verify}(pk, m, \sigma) \rightarrow \{1, 0\}$: is a deterministic verification algorithm that, given the public key pk, a message m and a signature σ, it outputs 1 if the signature verifies for me, and 0 if it does not verify.

Correctness Property [28]: For all messages m and all keys pairs (pk, sk), it holds $Verify(pk, m, Sign(sk, m)) = 1$.

The main advantages of using digital signatures are mentioned below [28]:

- They are publicly verifiable, *i.e.,* if Bob can verify that the message has been sent by Alice, then all other parties will also verify that message.
- Signatures are transferable, *i.e.,* Bob who can verify the signature can copy the signature and convince other parties that Alice is the author of that message.
- Digital signatures provide **non-repudiation**, *i.e.,* if Alice signs a message, she cannot later lie that she has not signed that message.

The Signal protocol uses Ed25519 digital signature scheme to provide signatures for keys [11].

3

Signal

Everyday people around the world use instant messaging apps to call, text or send photos and videos to each other. Many of us believe that our information is secret and nobody can get it without our permission. How can one say with confidence that his/her information cannot be stolen? For example, if Alice sends a message to Bob, can she be sure that nobody can steal the message or somehow get access to read the message, as depicted in Figure 3.1. Alice, in order being able to say that her way of messaging is safe and secure, she needs to have proper knowledge of how the app works and what kind of features and security properties it has to secure the communication.

Many messaging applications, such as WhatsApp, Facebook Messenger, Signal use the Signal protocol in order to achieve security properties. Therefore, it is essential to know how this protocol works. In this chapter, the construction and the working principles of the Signal protocol are explained.

Suggestions for reading. This Chapter is divided into 5 sections. Section 3.1 contains the general explanation of the Signal protocol where there is no information about the keys and mathematical computations that parties use to communicate. Section 3.2 contains the explanation of the Signal protocol with the notation in paper [11]. Section 3.3 contains the explanation of the Signal protocol with an easier notation, main objective of this thesis. Section 3.4 contains the information about the Signal app. Section 3.5 contains explanation how the Signal app and the Signal protocol work. The readers may choose one of Sections 3.2 or 3.3 since the content of the sections are the same and differ only in the notations. The notation in paper [11] is complicated and may confuse the reader. Therefore, it might be difficult to understand the mathematical calculations in Section 3.2. The notation in Section 3.3 is easier and more understandable for an ordinary reader.

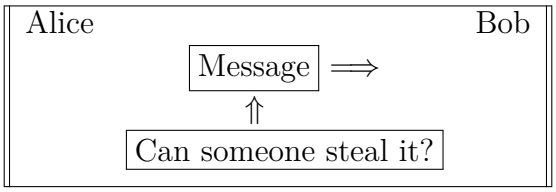

Figure 3.1: Can a message be stolen?

3.1 Overview of the Protocol

In this section, we discuss how the Signal protocol works for two parties, and we assume that Alice and Bob are willing to communicate over the Internet using the Signal App that uses the Signal protocol.

The Signal protocol establishes an **end-to-end encryption** between two users. In simple words this means that Alice and Bob are the only one who can read the messages of their communication chat.

The Signal protocol can be divided into 4 phases as explained in the following resources [11, 55, 56]:

1. **Registration phase:** if Alice wants to send a message to Bob, she needs to have some preknowledge about Bob's data. This to achieve and start a secure communication. How can they get each others information? Is there any trusted entity that will keep their public information secret? The answer to these questions is yes: Signal relies a trusted entity that holds public information connected to all parties that are registered in the system. This third party is called Key Distribution System (KDS). Parties that want to communicate send their information and get registration in KDS. Therefore, both Alice and Bob send their public information to KDS in order to register themselves with the system as depicted in Figure 3.2.

   When one wants to send a message or receive messages from others using specific instant messaging app, he/she need to register in that app. It is a part of the registration procedure to collect the public information to be registered so that one can message to other party using this information. For example, all people have an identity that is registered in the app and helps them to identify themselves across the app. So, if someone wants to send a message to other person, he/she needs to click on that person's username in order to start chatting. So, the user name is a part of the public information which is used to communicate to other people.
2. **Session setup:** in this phase, different public keys and shared secrets are computed. Alice and Bob should have a shared secret that is known only to them. Once Alice receives Bob's public information from the KDS, she generates keys that are needed to communicate with Bob. After she sends the message with associated information to Bob, he uses that information to check if the received information matches the data he has. If yes, then he can generate the same keys as Alice and therefore a shared key with Alice is computed as depicted in Figure 3.2.

   Let us explain it in an easier way. Alice and Bob have already registered in the app and they have each other in their contact list. If Alice want to send a message to Bob, she just needs to click on Bob's username and start the chat. When Alice attempts to do so, she retrieves Bob's public information and in the background of the app, the shared secrets is calculated based on it. The same is calculated for Bob. Now Alice and Bob can send messages to each other [50].
3. **Symmetric-ratchet communication:** in this phase, symmetric keys are

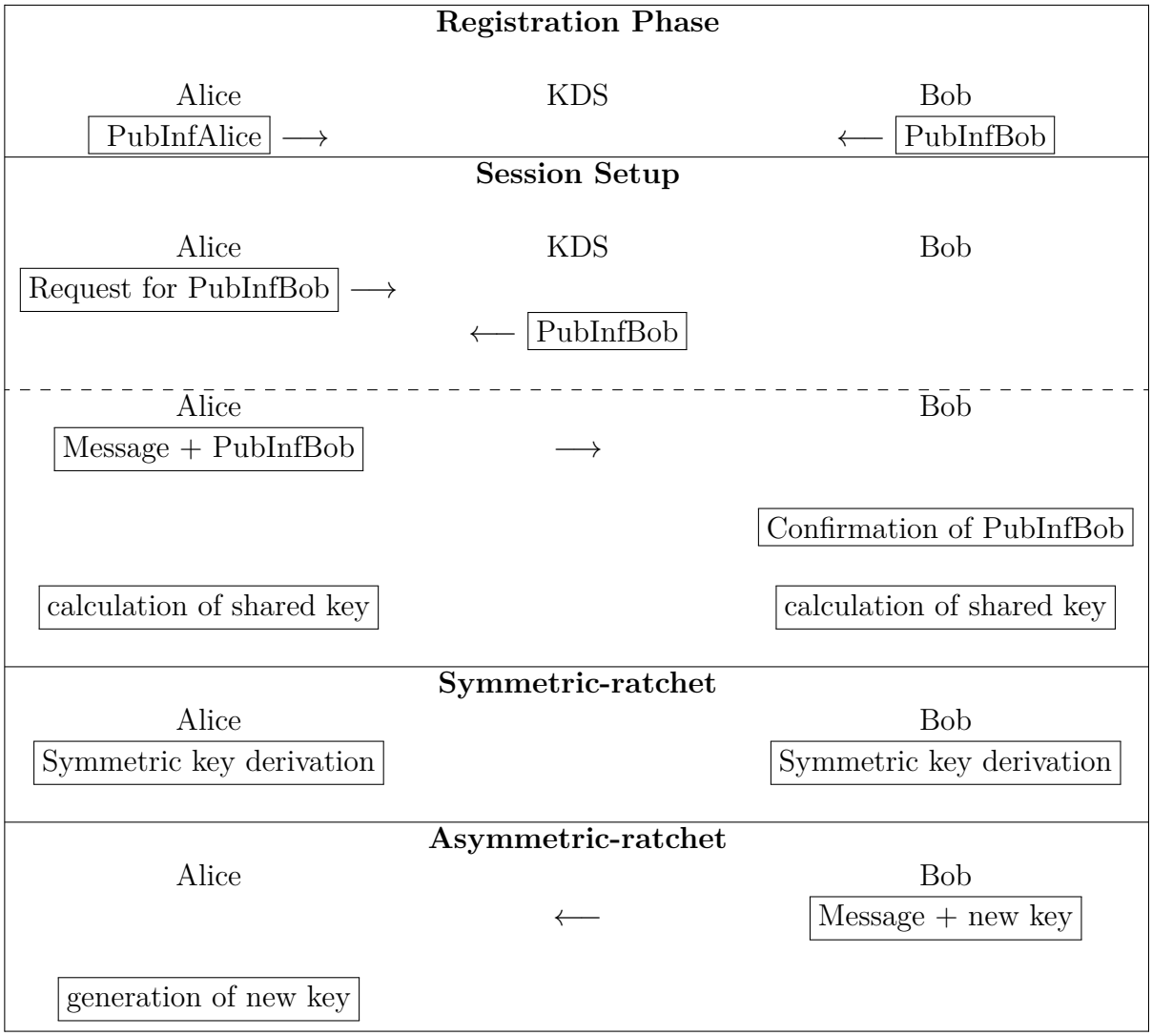

Figure 3.2: The overview of the Signal protocol.

derived. Some of these keys will be used to encrypt and decrypt the messages during the communication and some will be needed to update the keys each time.

When Alice receives the message from Bob, she checks the keys that comes with the message. If Alice does not have the corresponding keys that are received with the message, she does an asymmetric-ratchet update as depicted in Figure 3.2. Otherwise, she just decrypts the message.

In other words, Alice and Bob own symmetric keys which help them to encrypt or decrypt the message. Let us consider symmetric keys $Keys_A$, owned by Alice, and $Keys_B$, owned by Bob, where $Keys_A = Keys_B$. If Bob send a message to Alice, he encrypts the message with key $Keys_B$. Then the message can be decrypted by Alice using key $Keys_A$. But before the decryption, Alice checks if she has that corresponding symmetric key to decrypt the message using the data that comes with the new message. If yes, she does the step explained above. Otherwise Alice does an symmetric-ratchet.

4. **Asymmetric-ratchet update:** if one of the parties receives a message with a new key, then it generates the new keys in order to be able to decrypt the message and continue the communication (see Table 3.2).

   If Alice receives a message that Bob encrypted with key Key_C and does not have the corresponding symmetric key to decrypt that message. She does an asymmetric-ratchet to derive a new key that will decrypt the message that has been encrypted with Key_C. Then using that key Alice will be able to decrypt the message.

3.2 Formal Explanation of the Signal Protocol

3.2.1 Notation

Keys: The keys are divided into two lists: **asymmetric** (Table 3.1) and **symmetric** (Table 3.2). Let us consider Alice and identify her with A. All her belonging keys will be denoted with her identity in subscript.

Private keys	Public keys	Key generation	Explanation
ik_A	ipk_A	ik_A, $ipk_A \xleftarrow{R} \mathbb{Z}_q$	Long-term identity key pair
$prek_A$	$prepk_A$	$prek_A$, $prepk_A \xleftarrow{R} \mathbb{Z}_q$	Medium-term prekey pair
$eprek_A$	$eprepk_A$	$eprek_A$, $eprepk_A \xleftarrow{R} \mathbb{Z}_q$	Ephemeral prekey pair
ek_A	epk_A	ek_A, $epk_A \xleftarrow{R} \mathbb{Z}_q$	Ephemeral key pair

Table 3.1: A's asymmetric Keys.

Key	Explanation
$ck_A^{sym-ir:x,y}$	y^{th} key in Alice's x^{th} send chain
$ck_A^{sym-ri:x,y}$	y^{th} key in Alice's x^{th} receive chain
$mk_A^{sym-ir:x,y}$	y^{th} message key in Alice's x^{th} send chain
$mk_A^{sym-ri:x,y}$	y^{th} message key in Alice's x^{th} receive chain
rk_A^x	Alice's x^{th} root key

Table 3.2: A's symmetric Keys.

We define different prekeys in Table 3.1. A prekey is a shared secret between two parties that allows to verify the data or derive other keys during the communication.

The notations **"-ri"** and **"-ir"** on the keys in Table 3.2 shows if Alice is the sender or the receiver. If there is **"-ri"** on the key, this means she is a receiver (responder). If there is **"-ir"** on the key, this means she is an sender (initiator).

In the Signal protocol a signature scheme is used out as $Sign_{k_1}(k_2)$ which means the key k_2 is signed using the key k_1. The Signature scheme used is **Ed25519** signature scheme [11].

Each of the communicating parties have several one-time prekeys and they might publish them in KDS. We use the sign $[, k]$ to show that there several number keys k that are published.

Key Chains. In the Signal protocol, keys are derived using a Key Derivation Function (KDF) chain. Assume the KDF to be a black box which takes as input a key and outputs a "derived" key. Part of its output key is used as input for the next KDF, as shown in Figure 3.3. We call it **KDF chain** [37].

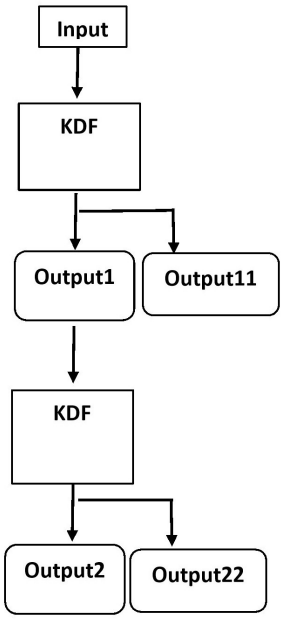

Figure 3.3: KDF chain workflow in Section 3.2.

In the Signal Protocol, there are two different chains [11]:

1. **Message chain update:** KDF_m is used for updating the message chain and it uses a chain key ck^i as input and outputs a new chain key ck^{i+1} and a message key mk^i.
2. **Root chain update:** KDF_r is used for updating the root chain and it takes as input the root key rk^i and a group element dh and outputs a new root key rk^{i+1} and chain key ck^{i+1}.

The key derivation functions are used in the Signal Protocol either **HMAC-SHA256** [34] or **HKDF-SH256** [21].

$asym - ri : x$ and $asym - ir : x$ are used as stage notations (see Section 3.2.2). The asymmetric ratchet is divided into two according whether a user wants to begin a receiving chain using a received ratchet or needs a new ratchet key which will be used to start a sending chain. In the first case, $asym - ir : x$ and in the second case, $asym - ri : x$ is used.

The notations $sym - ri : x, y$ and $sym - ir : x, y$ on the keys in Table 3.2 shows the y^{th} symmetric key on the x^{th} receive and send chain, respectively. These no-

tations are also used as stages where the numbers of updates in the x^{th} symmetric chain are counted in y. In other words, since each symmetric chain start after the asymmetric ratchet calculations are performed, we count the number of the symmetric chains that are initialized after the asymmetric ratchet using the notations $sym-ri:x,y$ and $sym-ir:x,y$. For instance, if we have a key with superscript $sym-ir:2,3$, this means the third symmetric key on the second asymmetric chain.

3.2.2 Four Phases of the Signal Protocol

In this subsection, each phase is explained with keys and computations that Alice and Bob use to communicate. The original notation [11] is used in this section.
Registration phase Both Alice and Bob register themselves to the KDS by providing the DH keys as depicted in Figure 3.4 [11]:

- a long-term public ik;
- a medium-term public signed $prek$;
- multiple short-term public $eprek$.

We will explain why above mentioned keys have different lifespan and why the different lifespan is important in Section 4.1.

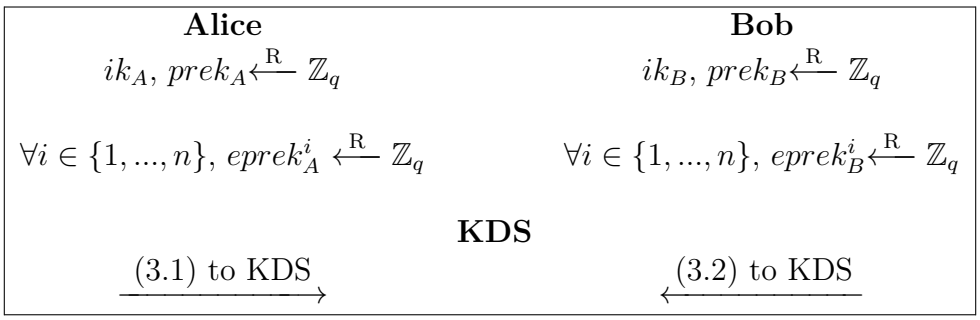

Figure 3.4: Registration phase in the Signal protocol.

***Calculation step by step* Alice:**

- Choose ik_A randomly from $\mathbb{Z}_q$;
- Choose $prek_A$ randomly from $\mathbb{Z}_q$;
- Choose multiple $eprek_A$ randomly from $\mathbb{Z}_q$;
- Send the following list of public keys to the KDS for registration:

$$ipk_A, prepk_A,\ Sign_{ik_A}(prepk_A),\ \{eprepk_A^i\}_{i=1}^n \tag{3.1}$$

Bob does the same and also sends to the key distribution server for registration:

$$ipk_B, prepk_B, Sign_{ik_B}(prepk_B),\ \{eprepk_B^i\}_{i=1}^n \tag{3.2}$$

Session setup phase. In this phase, the Signal Key Exchange Protocol is used, and different DH public keys and shared secrets are computed in order to initialize the session (see Figure 3.5).

Since Signal uses prekeys, it allows it to be an asynchronous protocol. This means by fetching one of the keys that is in KDS in the form of a Diffie-Hellman

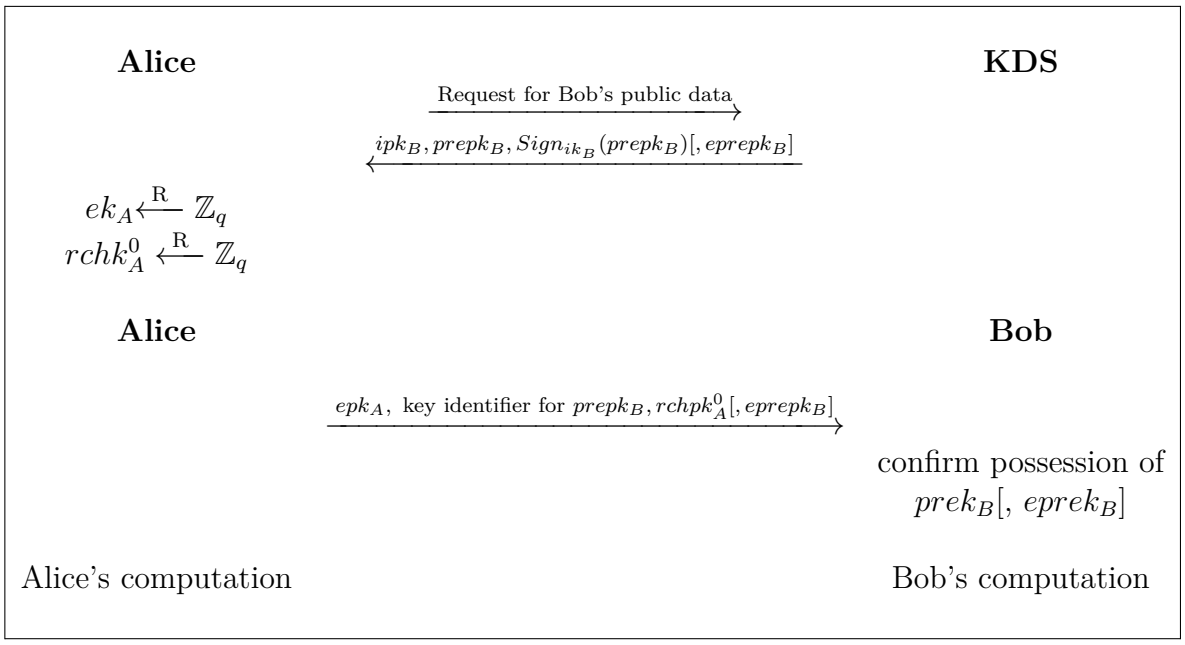

Figure 3.5: Session setup in the Signal protocol.

ephemeral public key, a session is established even if the parties are offline. The parties also share a medium-term key. This process allows a session to continue even if one-time ephemeral keys are compromised [11, 56].

The session setup phase begins with an initial stage called **receiving ephemerals** in which Alice requests the KDS for Bob's public keys and use them to establish a secure communication channel with him. Then she is provided with keys: Bob's identity public key ipk_B, Bob's current signed prekey $prepk_B$ and Bob's one-time prekey $eprepk_B$.

Parties that want to send a message to Bob fetch $prepk_B$ which is the medium-term key. Once used, the one-time key is deleted by the key directory from Bob's one-time prekey list.

After obtaining Bob's keys Alice starts building a session by generating her ephemeral key ek_A, and computes a session key. Alice concatenates these results and derives the initial root key rk^1 which is used to derive chain keys and sending chain key $ck^{sym-ir:0,0}$using a key derivation function. Finally, Alice also derives a ratchet key $rchk_A^0$using own keys and Bob's keys combination. She sends an initial message which contains the identifier of the prekey used so that Bob knows which prekey was used.

Bob receives the message from Alice and checks if the private keys known by him matches the public keys that Alice used. If the private keys correspond to the public keys that are sent by Alice, then Bob derives the same root key rk^1 and chain key $ck^{sym-ir:0,0}$ as Alice [11].

Calculations step by step Above we explained generally, what happens in Alice's and Bob's side (Table 3.4 and 3.3). Now we will show you how they make mathematical calculations. In this calculations, we assume that both Alice and Bob

Alice's computations
$ms \leftarrow (prepk_B)^{ik_A} \mid\mid (ipk_B)^{ek_A} \mid\mid (eprek_B)^{ek_A}$
if $eprepk_B$ then $ms \leftarrow ms \mid\mid (eprepk_B)^{ek_A}$
$rk^1,\ ck^{sym-ir:0,0} \leftarrow KDF_r(ms)$
$ck^{sym-ir:0,1},\ mk^{sym-ir:0,0} \leftarrow KDF_m(ck^{sym-ir:0,0})$

Table 3.3: Computation made by Alice in session setup phase of the Signal protocol.

Bob's computations
$ms \leftarrow (ipk_A)^{prek_B} \mid\mid (epk_A)^{ik_B} \mid\mid (epk_A)^{prek_B}$
if $eprepk_B$ then $ms \leftarrow ms \mid\mid (epk_A)^{eprek_B}$
$rk^1,\ ck^{sym-ir:0,0} \leftarrow KDF_r(ms)$
$ck^{sym-ir:0,1},\ mk^{sym-ir:0,0} \leftarrow KDF_m(ck^{sym-ir:0,0})$
$rchk_B^0 \xleftarrow{R} \mathbb{Z}_q$

Table 3.4: Computation made by Bob in session setup phase of the Signal Protocol.

have already registered themselves with the key distribution server and Alice wants to send a message to Bob.

1. Alice gets ipk_B, $prepk_B$, $Sign_{ik_B}(prepk_B)$ and $eprepk_B$ from the KDS;
2. Alice chooses ek_A randomly from $\mathbb{Z}_q$;
3. Alice chooses $rchk_A^0$ randomly from $\mathbb{Z}_q$;
4. Alice sends initial message to Bob where she attaches epk_A, the key identifier for $eprepk_B^i$, $rchpk_A^0$[,$eprepk_B$];
5. Bob checks if he has the corresponding private keys for $prek_B$[, $eprek_B$];
6. In this stage, Alice and Bob calculate a master secret ms and some necessary keys that will be used further in messaging as shown in Table 3.3:

Ratchet. The Double Ratchet is core part of the Signal protocol that provides the encrypted messages exchange. The Double ratchet contains two phases of the Signal protocol called **symmetric-ratchet** (Figure 3.6) and **asymmetric-ratchet** (Figure 3.7). We will explain how the ratchet algorithm works and what kind of calculations are done in each phase.

Symmetric-ratchet phase. In this phase, **receiving** and **sending** symmetric keys are derived. Alice generates an updated sending chain key and a sending message key by using the current sending chain key. Alice uses the sending message key to encrypt the message that she wants to send to Bob. This message key is unique for each message. Then she deletes the sending message key and the old sending chain key [11, 37, 55, 56].
The calculations are explained below:

- Alice sends a message m to Bob in stage $[sym - ir : x, y]$. She calculates and sends the following:

$$[AEAD_{mk^{sym-ir:x,(y-1)}}(m, AD = \emptyset), rchpk_A^x, ipk_A, ipk_B, y] \tag{3.3}$$

- Then both Alice and Bob calculate the next chain key and message key:

$$ck^{sym-ir:x,(y+1)}, mk^{sym-ir:x,y} \leftarrow KDF_m(ck^{sym-ri:x,y}) \tag{3.4}$$

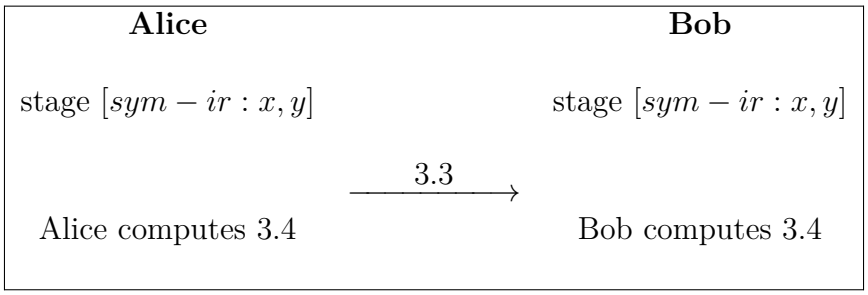

Figure 3.6: Symmetric-ratchet phase in the Signal protocol.

After Alice receives a message from Bob, the first step is checking the ratchet key contained in the ciphertext, i.e., $rchpk_B^x$. If Alice does not have the corresponding ratchet key, she does the asymmetric-ratchet update. Otherwise, she takes the index of the message and selects the receiving chain which is indicated by the index and then derives the corresponding receiving message key. She uses the new receiving message key to decrypt the message.

Asymmetric-ratchet phase. In this step, the chain keys are updated using Diffie-Hellman outputs. This update is executed when Alice receives a message with a new ratchet public key. There are two steps that Alice follows [11, 37, 55, 56]:

1. She uses two ratchet key combination in order to compute a DH shared secret: the ratchet public key that she received from Bob and her old ratchet private key. Then Alice combines the computed value with her root chain key and generates a receiving chain key and receiving message key. The calculations are explained below:
   - Bob calculates the following keys in stage $[asym - ri : x]$:

$$tmp, ck^{sym-ri:x,0} \leftarrow KDF_r(rk^x, (rchpk_A^{(x-1)})^{rchk_B^{(x-1)}}) \tag{3.5}$$

$$ck^{sym-ri:x,1}, mk^{sym-ri:x,0} \leftarrow KDF_m(ck^{sym-ri:x,0}) \tag{3.6}$$

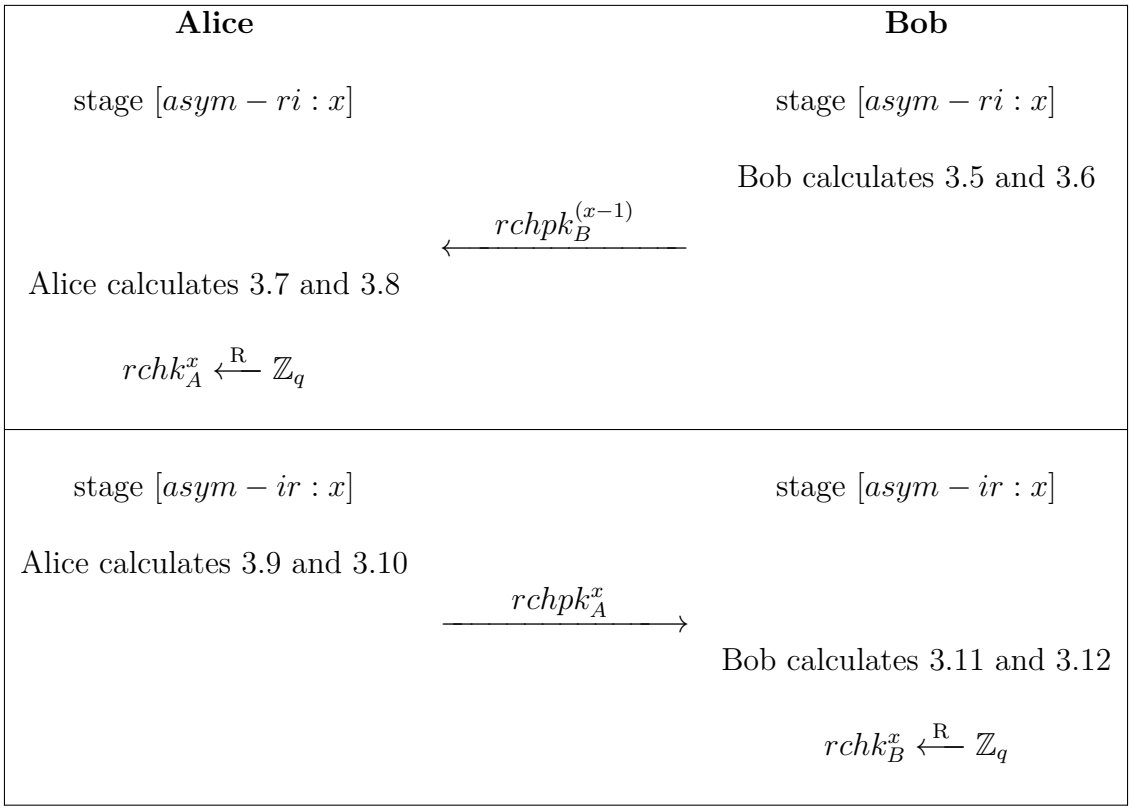

Figure 3.7: Asymmetric-ratchet phase in the Signal protocol.

- Then Bob sends $rchpk_B^{(x-1)}$ to Alice;
- Alice also calculates the following keys in stage $[asym-ri : x]$ and chooses $rchpk_A^x$ randomly from $\mathbb{Z}_q$:

$$tmp, ck^{sym-ri:x,0} \leftarrow KDF_r(rk^x, rchpk_B^{(x-1)}) \tag{3.7}$$

$$ck^{sym-ri:x,1}, mk^{sym-ri:x,0} \leftarrow KDF_m(ck^{sym-ri:x,0}) \tag{3.8}$$

2. Alice uses two ratchet key combination in order to compute a second DH shared secret: the ratchet public key that she received from Bob and her new ratchet private key. Then the derived value is combined with the root chain key and the DH shared secret that is derived in the first step in order to generate a new sending chain key, sending message key and the root chain key. The calculations are explained below:

- Alice calculates the following keys in stage $[asym - ir : x]$:

$$rk^{x+1}, ck^{sym-ri:x,0} \leftarrow KDF_r(tmp, (rchpk_B^{(x-1)})^{rchk_A^x}) \tag{3.9}$$

$$ck^{sym-ri:x,1}, mk^{sym-ir:x,0} \leftarrow KDF_m(ck^{sym-ri:x,0}) \tag{3.10}$$

- Then Alice sends $rchpk_A^x$ to Bob;

- Bob also calculates the following keys in stage $[asym - ir : x]$:

$$rk^{x+1}, ck^{sym-ri:x,0} \leftarrow KDF_r(tmp, (rchpk_A^x)^{rchk_B^{(x-1)}}) \tag{3.11}$$

$$ck^{sym-ri:x,1}, mk^{sym-ir:x,0} \leftarrow KDF_m(ck^{sym-ri:x,0}) \tag{3.12}$$

  Chooses $rchk_B^x$ randomly from $\mathbb{Z}_q$.

3.3 Explanation with Easy Notations

3.3.1 New Notation

Keys: The keys are divided into two lists: **asymmetric** (Table 3.5) and **symmetric** (Table 3.6). Let us consider Alice and identify her with A. All her belonging keys will be denoted with her identity in subscript.

The new notation makes easier to identify the key: if it has a superscript, it is a symmetric key and the first letters will characterize if it is a receiving/sending key and if it is a message/chain key. The same applies for the asymmetric keys in which the first letter always identify if the key is the identity, pre, one-time, ephemeral key. For example, the keys $ck_A^{sym:-ir:x,y}$ and $ck_A^{sym:-ri:x,y}$ are now identified as $sck_A^{x,y}$ and $rck_A^{x,y}$

Private keys	Public keys	Key generation	Explanation
ik_A	ipk_A	ik_A, $ipk_A \xleftarrow{R} \mathbb{Z}_q$	Long-term identity key pair
pk_A	ppk_A	pk_A, $ppk_A \xleftarrow{R} \mathbb{Z}_q$	Medium-term prekey pair
ok_A	opk_A	ok_A, $opk_A \xleftarrow{R} \mathbb{Z}_q$	Ephemeral "one-time prekey" pair
ek_A	epk_A	ek_A, $epk_A \xleftarrow{R} \mathbb{Z}_q$	Ephemeral key pair

Table 3.5: A's new asymmetric keys.

Key	Explanation
$sck_A^{\{x,y\}}$	y^{th} key in Alice's x^{th} send chain
$rck_A^{\{x,y\}}$	y^{th} key in Alice's x^{th} receive chain
$smk_A^{\{x,y\}}$	y^{th} message key in Alice's x^{th} send chain
$rmk_A^{\{x,y\}}$	y^{th} message key in Alice's x^{th} receive chain
rk_A^x	Alice's x^{th} root key

Table 3.6: A's new symmetric keys.

We define different prekeys in Table 3.5. Prekey is a shared secret between two parties that allows to verify the data or derive other keys during the communication.

In the Signal protocol a signature scheme is used out as $Sign_{k_1}(k_2)$ which means the key k_2 is signed using the key k_1. The Signature scheme used is **Ed25519** signature scheme [11].

Each of the communicating parties have several one-time prekeys and they might publish them in KDS. We use the sign $[, k]$ to show that there several number keys k that are published.

Key Chains. In the Signal protocol, keys are derived using a KDF chain. Assume the KDF is a black box and it takes some input data. Some part of its output data used as an input data for the next step. Since the next step uses the output of the previous step in order to derive new output as shown in Figure 3.8. We call it **KDF chain** [37].

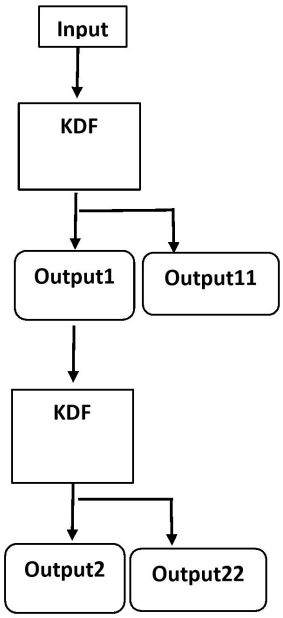

Figure 3.8: KDF chain workflow in Section 3.3.

There are two chain updates as stated in [11]:

1. **Message chain update:** KDF_m is used for message chain update and it takes as an input chain key ck^i and outputs a new chain key ck^{i+1} and a message key mk^i.
2. **Root chain update:** KDF_r is used for root chain update and it takes as and input rk^i and DH value dh and outputs a new root key rk^{i+1} and chain key ck^{i+1}.

Here the key derivation functions are either **HMAC-SHA256** [34] or **HKDF-SH256** [21].

Asymmetric ratchet is divided into two according whether a user wants to begin a receiving chain using a received ratchet or generates a new ratchet key in order to begin a sending chain. The notation $\{x, y\}$ on the keys in Table 3.6 shows the y^{th} symmetric key on the x^{th} receive or send chain. In other words, since each symmetric chain start after the asymmetric ratchet calculations are performed, we count the number of the symmetric chains that are initialized after the asymmetric ratchet

using the notation $\{x, y\}$. For instance, if we have a key with a notation $\{2, 3\}$ on the superscript, this means the third symmetric key on the second asymmetric chain.

3.3.2 Four Phases of the Signal Protocol with the New Notations

In this subsection, each phase is explained with keys and computations that Alice and Bob use to communicate. Consider the fact that the notations that are used in this section are the contribution of the thesis.

Registration phase. Both Alice and Bob register themselves to the KDS by providing the DH keys as depicted in Figure 3.9):

- a long-term public ik;
- a medium-term public signed prekey pk;
- multiple short-term public ok.

We will explain why above mentioned keys have different lifespan and why the different lifespan is important in Section 4.1.

***Calculation step by step* Alice:**

- Choose ik_A randomly from $\mathbb{Z}_q$;
- Choose pk_A randomly from $\mathbb{Z}_q$;
- Choose multiple ok_A randomly from $\mathbb{Z}_q$;
- Send the following list of public keys to the KDS for registration:

$$ipk_A, ppk_A,\ Sign_{ik_A}(ppk_A),\ \{opk_A^i\}_{i=1}^n \tag{3.13}$$

Bob does the same and also sends to the key distribution server for registration:

$$ipk_B, ppk_B, Sign_{ik_B}(ppk_B),\ \{opk_B^i\}_{i=1}^n \tag{3.14}$$

The notions explained above are illustrated in Figure 3.9.

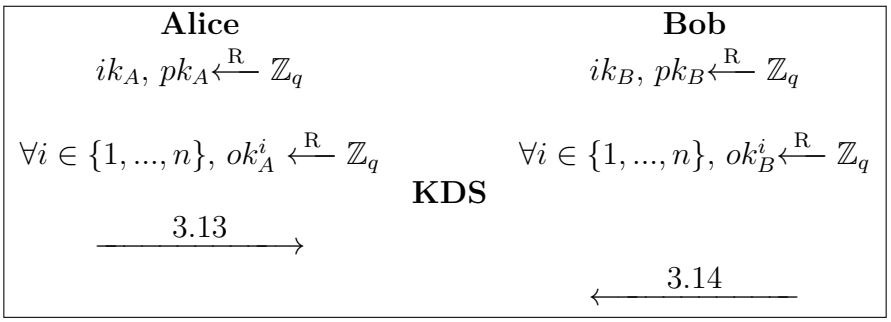

Figure 3.9: Registration phase in the Signal protocol explained with the new notation.

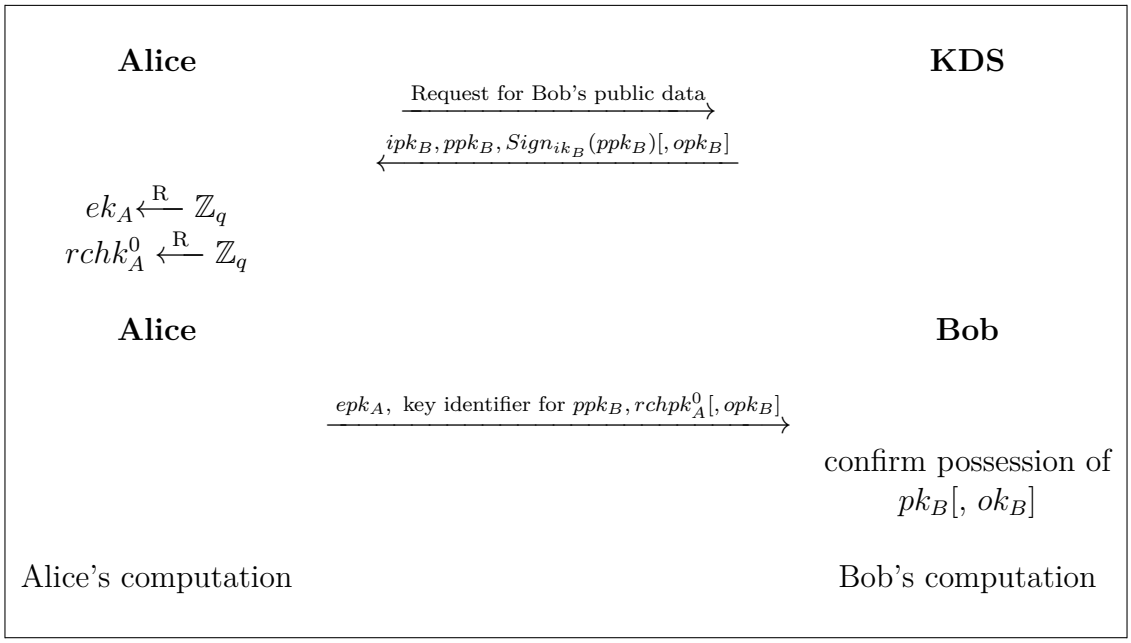

Figure 3.10: Session setup in the Signal protocol explained with the new notations.

Session setup phase. In this phase, the Signal Key Exchange Protocol is used, and different DH public keys and shared secrets are computed in order to initialize the session (Figure 3.10).

Since Signal uses prekeys, it allows it to be an asynchronous protocol. This means by fetching one of the keys that is in the KDS in the form of a Diffie-Hellman ephemeral public key, a session is established even if the parties are offline. The parties also share a medium-term key. This process allows a session to continue even if one-time ephemeral keys are compromised [11, 56].

The session setup phase begins with an initial stage called **receiving ephemerals** in which Alice requests the KDS for Bob's public keys and use them to establish a secure communication channel with him. She receives the identity public key ipk_B, the current signed prekey ppk_B and a one-time prekey opk_B.

Parties who wants to send a message to Bob fetch ppk_B which is the medium-term key. Once used, the one-time key is deleted by the key directory from Bob's one-time prekey list .

After obtaining Bob's keys Alice starts building a session by generating her ephemeral key ek_A, and computes a session key. Alice concatenates these results and derives the initial root key rk^1 and sending chain key $sck^{0,0}$ using a key derivation function. Finally, she also generates an ephemeral DH key called ratchet key $rchk_A^0$. She sends an initial message which contains the identifier of the prekey used so that Bob knows which prekey was used.

Bob receives the message from Alice and checks if the private keys known by him matches the public keys that Alice used. If the private keys corresponds to the public keys that are sent by Alice, then Bob derives the same root key rk^1 and chain key $sck^{0,0}$ as Alice [11].

Calculations step by step Above we explained generally, what happens in Alice's and Bob's side (Table 3.7 and 3.8). Now we will show you how they make mathematical calculations. In this calculations, we assume that both Alice and Bob have already registered themselves with the key distribution server and Alice wants to send a message to Bob.

Alice's computations
$secret \leftarrow (ppk_B)^{ik_A} \mid\mid (ipk_B)^{ek_A} \mid\mid (ok_B)^{ek_A}$
if opk_B then $secret \leftarrow secret \mid\mid (opk_B)^{ek_A}$
$rk^1,\ sck_A^{0,0} \leftarrow KDF_r(secret)$
$sck_A^{0,1},\ smk_A^{0,0} \leftarrow KDF_m(sck_A^{0,0})$

Table 3.7: Computation made by Alice in Session Setup Phase of Signal Protocol Using the New Notations.

1. Alice gets ipk_B, ppk_B, $Sign_{ik_B}(ppk_B)$ and $eprepk_B$ from the KDS;
2. Alice chooses ek_A randomly from $\mathbb{Z}_q$;
3. Alice chooses $rchk_A^0$ randomly from $\mathbb{Z}_q$;
4. Alice sends initial message to Bob where she attaches epk_A, the key identifier for $eprepk_B^i$, $rchpk_A^0$[,opk_B];
5. Bob checks if he has the corresponding private keys for pk_B[, ok_B];
6. In this stage, Alice and Bob calculate shared secret and some necessary keys that will be used further in messaging.

Ratchet. The Signal protocol uses the Double ratchet to exchange the encrypted messages. The Double ratchet contains two phases of the Signal protocol called **symmetric-ratchet** (Figure 3.11) and **asymmetric-ratchet** (Figure 3.12) phases. We will explain how the ratchet algorithm works and what kind of calculations are done in each phase.

Symmetric-ratchet phase. In this phase, **receiving** and **sending** symmetric keys are derived. Alice generates an updated sending chain key and a sending message key by using the current sending chain key. Alice uses the sending message key to

Bob's computations
$secret \leftarrow (ipk_A)^{pk_B} \mid\mid (epk_A)^{ik_B} \mid\mid (epk_A)^{pk_B}$
if opk_B then secret$\leftarrow$ secret $\mid\mid$ $(epk_A)^{ok_B}$
$rk^1,\ sck_B^{0,0} \leftarrow KDF_r(secret)$
$sck_B^{0,1},\ smk_B^{0,0} \leftarrow KDF_m(sck_B^{0,0})$
$rchk_B^0 \xleftarrow{R} \mathbb{Z}_q$

Table 3.8: Computation made by Bob in Session Setup Phase of Signal Protocol Using the New Notations.

encrypt the message that she wants to send to Bob. This message key is unique for each message. Then she deletes the sending message key and the old sending chain key [11, 37, 55, 56].
The calculations are explained below:

- Alice sends a message m to Bob. She calculates and sends the following:

$$AEAD_{smk^{x,y-1}}(m, AD = \emptyset), rchpk_A^x, ipk_A, ipk_B, y \tag{3.15}$$

- Then both Alice and Bob calculate the next chain key and message key:

$$sck^{x,(y+1)}, smk^{x,y} \leftarrow KDF_m(sck^{x,y}) \tag{3.16}$$

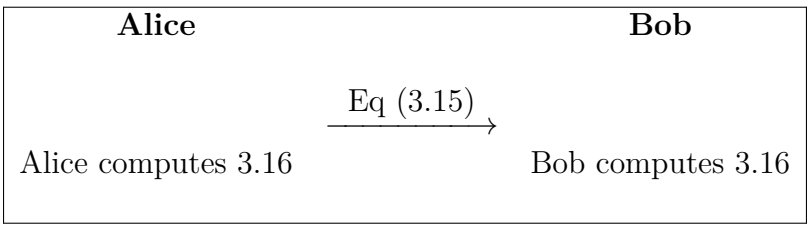

Figure 3.11: Symmetric-ratchet phase in the Signal protocol explained with the new notation.

When Alice receives an encrypted message from Bob, she checks the ratchet key contained in the ciphertext, i.e., $rchpk_B^x$. If Alice does not have the corresponding ratchet key, she does the asymmetric-ratchet update. Otherwise, she takes the index of the message and selects the specific receiving chain and she derives the corresponding receiving message key. She uses the new receiving message key to decrypt the message.

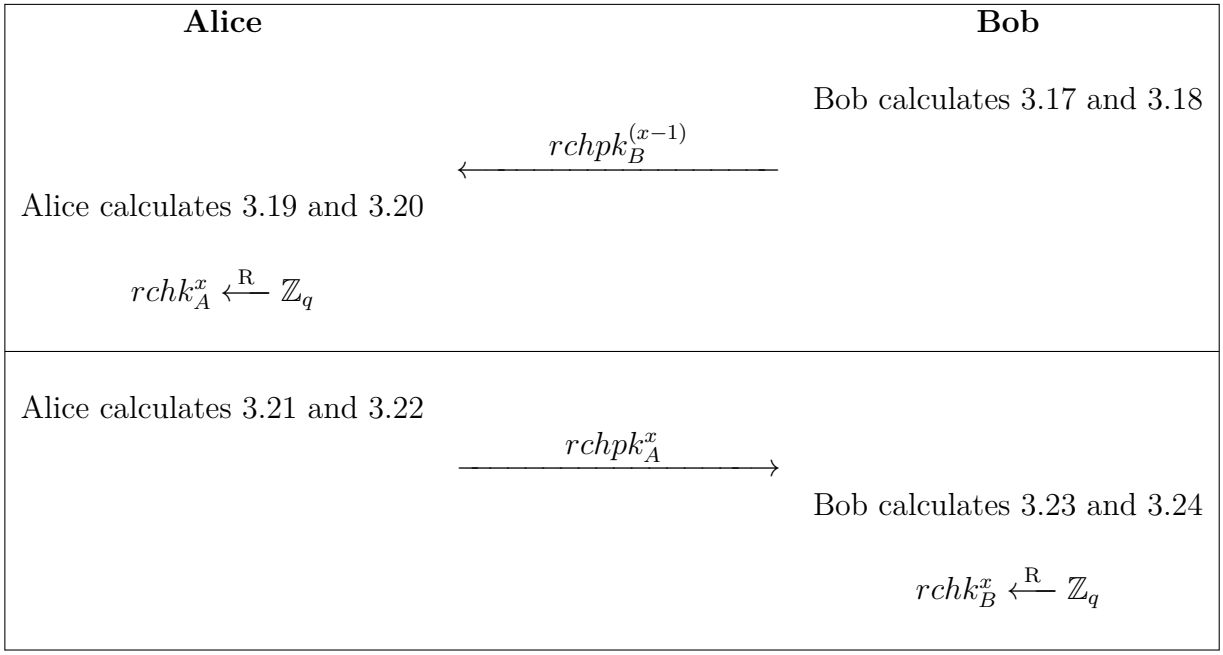

Figure 3.12: Asymmetric-ratchet phase in the Signal protocol explained with the new notation.

Asymmetric-ratchet phase. In this step, the chain keys are updated using Diffie-Hellman outputs.This update is executed when Alice receives a message with a new ratchet public key. There are two steps that Alice follows as described in [11, 37, 55, 56]:

1. She uses her received ratchet public key and her old ratchet private key to compute a DH shared secret. Then Alice combines the computed value with her root chain key and derives a receiving chain key and receiving message key. The calculations are explained below:
   - Bob calculates the following keys:

$$tmp, rck^{x,0} \leftarrow KDF_r(rk^x, (rchpk_A^{(x-1)})^{rchk_B^{(x-1)}}) \tag{3.17}$$

$$rck^{x,1}, rmk^{x,0} \leftarrow KDF_m(rck^{x,0}) \tag{3.18}$$

   - Then Bob sends $rchpk_B^{(x-1)}$ to Alice;
   - Alice also calculates the following keys and chooses $rchpk_A^x$ randomly from $\mathbb{Z}_q$:

$$tmp, rck^{x,0} \leftarrow KDF_r(rk^x, rchpk_B^{(x-1)}) \tag{3.19}$$

$$rck^{x,1}, rmk^{x,0} \leftarrow KDF_m(rck^{x,0}) \tag{3.20}$$

2. Alice uses the received ratchet public key and her new ratchet private key to compute a second DH shared secret. Then she combines it with the root chain key and the previously computed DH shared secret to derive a new sending

chain key, sending message key and the root chain key. The calculations are explained below:

- Alice calculates the following keys:

$$rk^{x+1}, sck^{x,0} \leftarrow KDF_r(tmp, (rchpk_B^{(x-1)})^{rchk_A^x}) \tag{3.21}$$

$$sck^{x,1}, smk^{x,0} \leftarrow KDF_m(sck^{x,0}) \tag{3.22}$$

- Then Alice sends $rchpk_A^x$ to Bob;
- Bob also calculates the following keys and chooses $rchk_B^x$ randomly from $\mathbb{Z}_q$.:

$$rk^{x+1}, sck^{x,0} \leftarrow KDF_r(tmp, (rchpk_A^x)^{rchk_B^{(x-1)}}) \tag{3.23}$$

$$sck^{x,1}, smk^{x,0} \leftarrow KDF_m(sck^{x,0}) \tag{3.24}$$

3.4 Signal Private Messenger

Have you ever tried to search for a messaging apps? If you tried perhaps the result for this request was screenshots, official pages and articles related to different messaging apps. All of them have in common the goal of *connecting people*.

Through out the thesis, Facebook Messenger and WhatsApp are mentioned as the most popular messaging apps that use the Signal protocol. There is also another app that uses the Signal protocol called **Signal**. This app is considered as one of the most secure messaging apps on the market at the moment. Therefore, it will be interesting to give information about the Signal app, how it works and explain how the Signal protocol is deployed in the Signal app. In this section, we just discuss what Signal is, advantages and limitation that it has.

Signal. Signal is an messaging app that provides encrypted communication. It is available for smartphones that use Android and iOS. It can also be used on platforms like Linux, Windows, and macOS.

Users can chat and make calls using the Signal app. Signal is like WhatsApp and Facebook Messenger but with focus on security and privacy. Alongside with security protocols, it also provides features such as group chat, video calls and emoji support that is available in other messaging apps.

Why use Signal? Signal uses **end-to-end encryption** which protects the communication from hackers and curious service providers. When Alice sends an encrypted message to Bob, first the message passes through the servers of the service provider and then Bob receives the message and he decrypts it.

If Alice and Bob communicate to each other, **end-to-end encryption** provides privacy between them, *i.e.,* only Alice and Bob can read the messages. End-to-end encryption does not allow even service providers to read the message since the message is encrypted through out the way it reaches Bob. It is not safe to use client server encryption since the message is decrypted by the servers of the service

provider and all the messages might be saved there. This means the service provider has access to the data that passes through their server. End-to-end encryption does not allow any third party to access the messages.

Signal also provides **mandatory encryption**. This means the messages are encrypted by default. There are messaging apps that provide end-to-end encryption as an option which means if one do not choose that option his/her messages will be encrypted using client server encryption.

Signal is an **open source project**, which makes it more reliable. Anyone can examine the code and see if it really has the features that is promised by the developers.

Signal is a **user friendly** app that provides security. It collects less information about the user. It just collects the data that is needed for the service to work [48].

Signal also provides **timers for messages**. User can set a timer to the messages and control when the messages can be deleted from sender's and receiver's device.

Limitations. On the other hand Signal has limitations as well.

The registration to Signal app may be a problem for some users. During registration, the user should use his/her phone number which means whoever wants to contact that user needs to have his/her phone number. Consider the following scenario. Someone wants to communicate securely and chooses Signal, but he/she does not want to share his/her number. In this case, it is impossible to use Signal.

The Signal app is not a famous instant messaging app. Therefore, there are not many people who use Signal. It is impossible to communicate using Signal if your contact does not use it.

3.5 The Signal App and the Signal Protocol

In this section, we will explain how the Android version of the Signal App interacts with the underlying Signal protocol, explained in Section 3.2 and 3.3. We will explain every single step using illustrations.

Every smartphone that works with Android has the **Play Store** where users can install apps that they need. It is enough to write *"Signal"*, choose the first option as presented in Figure 3.13 and install it. After installation, when the app is opened, users see the following request presented in Figure 3.14 on the screen of the smartphone.
After the user presses continue, he/she will get other three questions for security reason as shown in Figure 3.15. This allows the user to choose which information he/she wants to be accessible by the application.

App Registration. After all primary steps are done the app asks the user to have registration with his/her phone number. For this request the user needs to fill his/her phone number and press *"Register"* button (Figure 3.16).
When the user presses the *"Register"* button, Signal sends another request shown

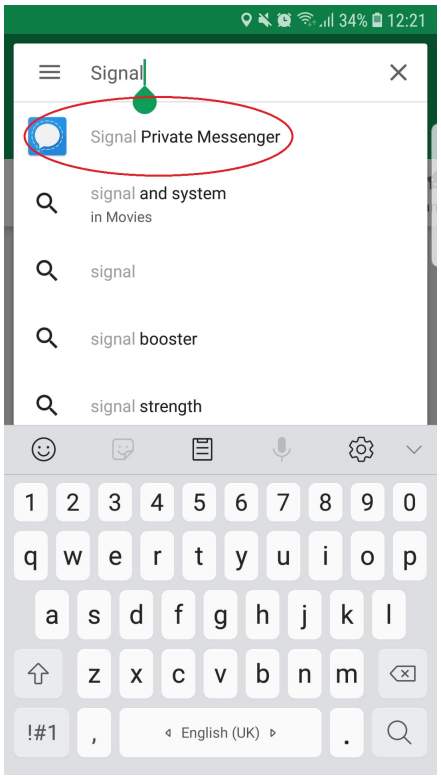

Figure 3.13: Searching the Signal Application in smartphone's "Play Store".

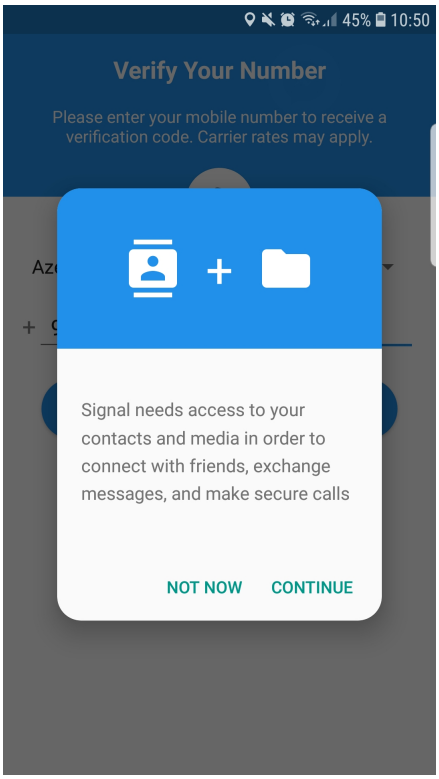

Figure 3.14: When Signal is installed and opened for the first time.

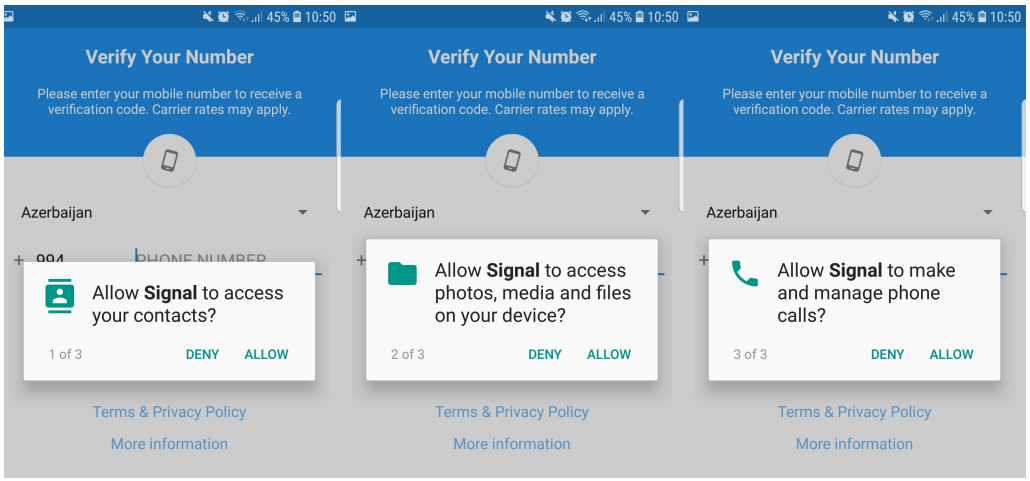

Figure 3.15: Signal asks to access user's data.

in Figure 3.17. If a user presses first the button *"Continue"* and then the button *"Allow"*, Signal will have an access to view SMSs. Now the user needs to verify himself/herself filling in the verification code that app sends to his/her phone number in the form of SMS (Figure 3.18).

If the user allowed Signal to view his/her SMSs, the verification code will be fetched

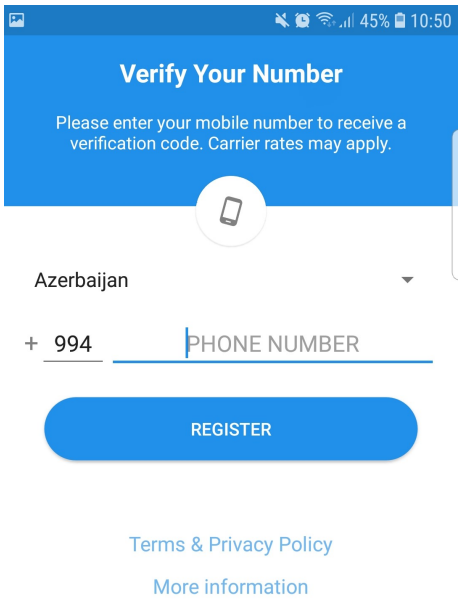

Figure 3.16: Registration phase in the Signal App.

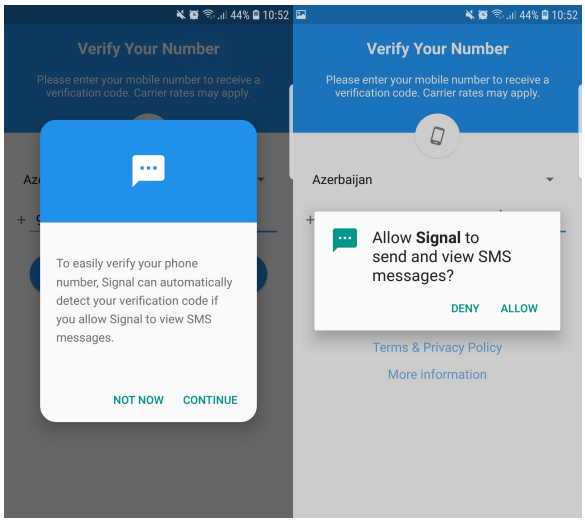

Figure 3.17: Signal asks to view SMS that is received on smatrphone.

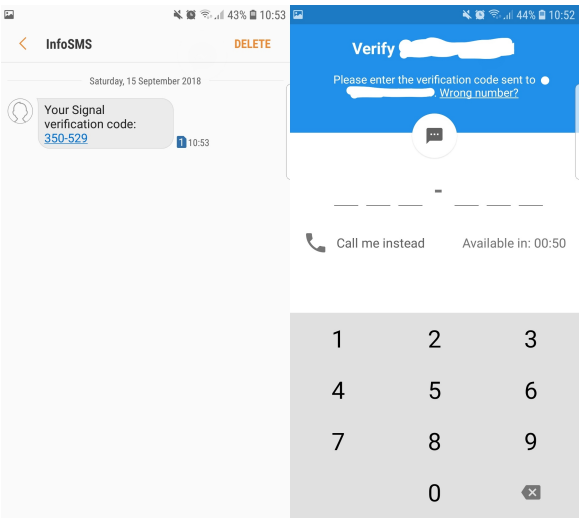

Figure 3.18: Verification of the Phone Number During the Registration Phase in The Signal App.

from the SMS and filled in the place as shown in Figure 3.18. Otherwise the user needs to fill the verification code manually.

When the user registers into the Signal app, the registered users information like phone number, public keys are stored on the server. This is the **registration phase** of the Signal protocol.

Starting to Chat. After all the above mentioned steps are done, the user sees the window on the screen that has an button on the right. When the user clicks on that, he/she goes to the window where the contacts are (Figure 3.19).

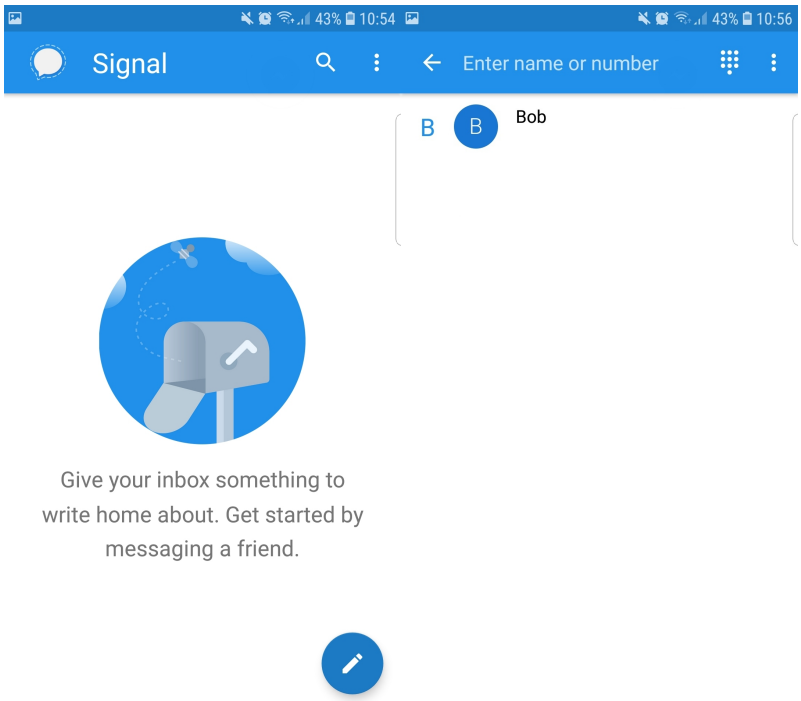

Figure 3.19: Start point of the conversation in the Signal App.

Now let us assume that Alice and Bob are messaging to each other. When Alice clicks on Bob's name as presented in Figure 3.19, the new page is opened and Alice can tap the message as shown in Figure 3.20. This is the **session setup phase** of the Signal Protocol where the shared secrets are computed according to the first message.

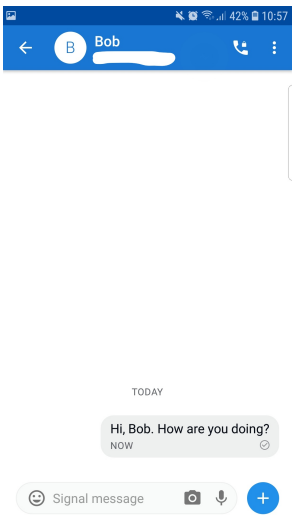

Figure 3.20: The first message sent using the Signal App.

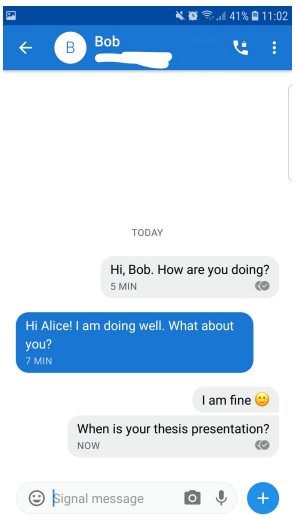

Figure 3.21: Alice sends message to Bob using the Signal App.

During the Conversation. When Alice sends messages to Bob, each message

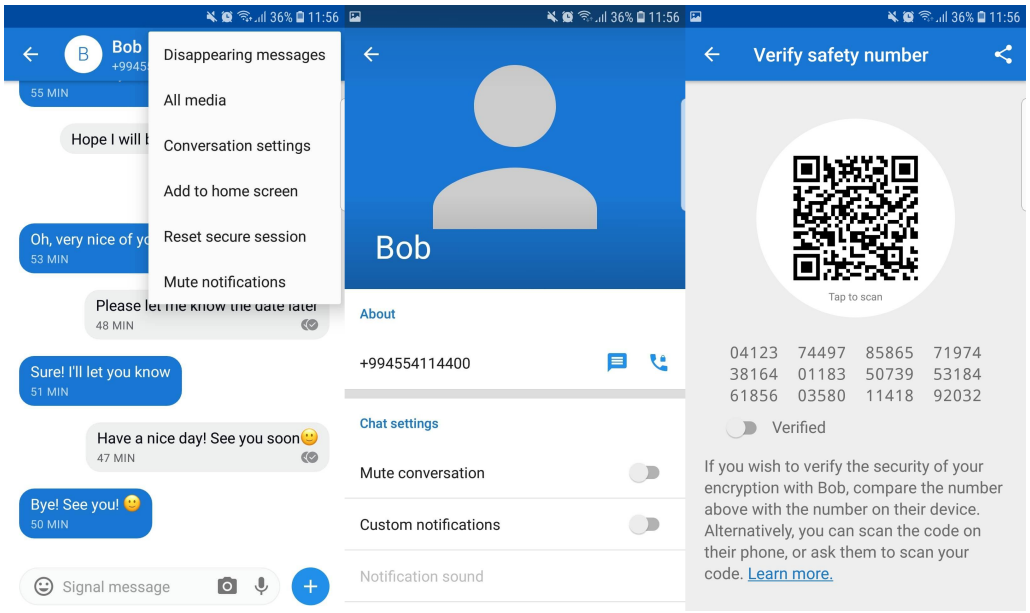

Figure 3.22: Safety numbers to control the security of the conversation in the Signal App.

is encrypted with a unique key. Each time when Alice sends new message, a new unique key is derived in order to encrypt the message. Alice cannot see how the keys are derived since this algorithm is implemented somewhere in the software that controls the app. This derivation of new keys is the **symmetric-ratchet phase** in Signal Protocol. Also each message is sent contains a specific data alongside with the message. This data says the receivers whether the received message is new or not. For Example, Alice sends a message to Bob and Bob gets offline. When he is online again, there can be several messages sent by Alice as shown in Figure 3.21. In this case, the software needs to derive new keys in order to decrypt and make visible the messages for Bob. This computation is the asymmetric-ratchet phase in Signal protocol and the specific data mentioned above is the ratchet key.

Some Details about Signal. There are some features that might be interesting for users.

If one sends a message to other person whose device is not connected to mobile data or Wi-Fi, on the right corner of the sent message appears a circle with one tick in it. If the person is connected to mobile data or wi-fi, on the right corner of the sent message appears a circle with two ticks in it. If the message is read, then that circle with two tick is shaded to grey.

Signal provides users with *"safety numbers"* in order to make sure that a user sends messages exactly to his/her contact's device. In order to see the safety numbers the user needs to do the steps as presented in Figure 3.22.

How do these safety numbers help the users? Assume Alice and Bob are messaging to each other. If they want to check and be sure that they are talking exactly to each other, they need to compare these safety numbers (Figure 3.23). Because the numbers are like a fingerprint of their shared secret. If the fingerprints are different,

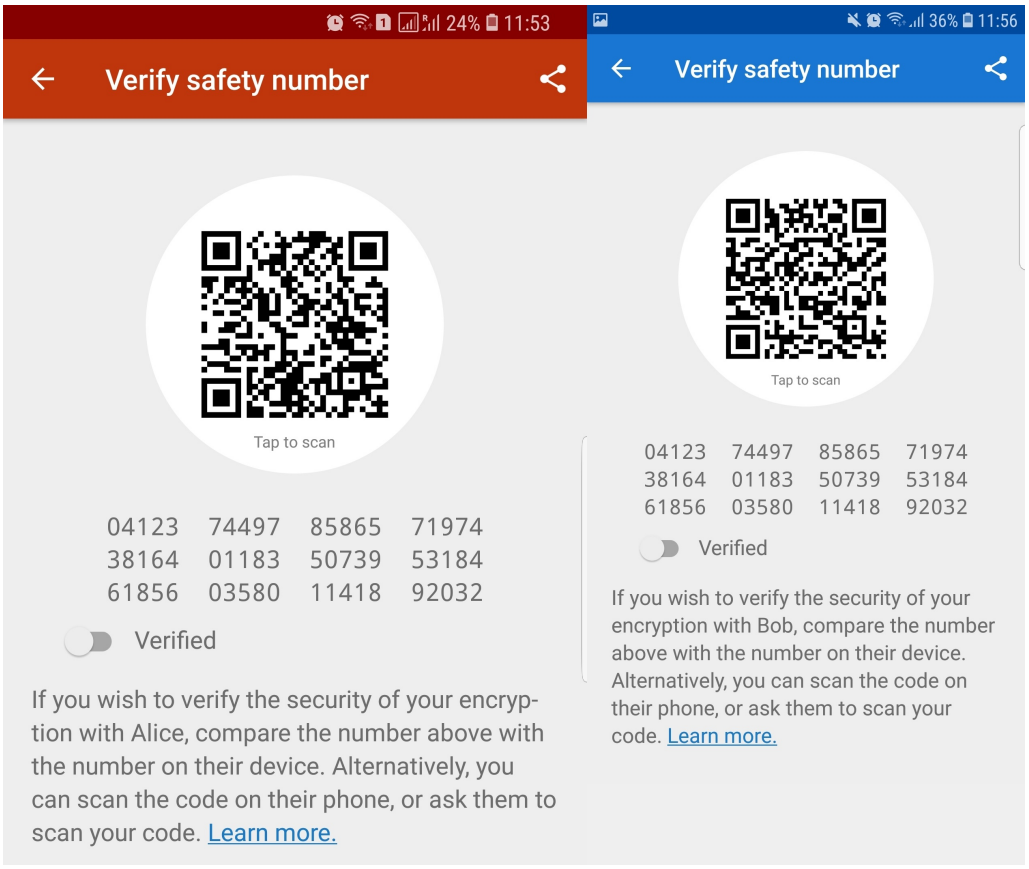

Figure 3.23: Comparing the Safety Numbers In the Signal App.

the protocol was not correctly executed and there might be something, or someone, odd.

4

Security Notions

The growth of computers, smartphones and messaging applications brings the security phenomenon into the first priority. The development of Cryptography as a science gives us a chance to prove the security of the algorithms and systems from the mathematical point of view. In sections 4.1 and 4.2, we will discuss the security notions that are relevant to Signal protocol namely: the forward secrecy, post-compromise security and the security parameters of AEAD. Also we will explain why the Signal protocol is secure by using the above mentioned notions.

4.1 Security Features

Perfect Forward Secrecy. When two parties communicate using the Signal protocol, they generate different key pairs such as a long-term identity key pair, a medium-term prekey pair, an ephemeral prekey pair and an ephemeral key. Now consider a situation when one of these key pairs or past session keys, is compromised which means that both private and public key are stolen. This case brings a question to our minds: can these two parties continue to communicate securely? Therefore, it is necessary to consider the following situations [1, 2]:

- long-term keys can be compromised;
- past session keys can be compromised.

Informally, a protocol is said to have **perfect forward secrecy** if compromise of long-term keys does not compromise past session keys such as medium-term prekeys, ephemeral prekeys, ephemeral keys, sending and receiving chain keys, message keys, root key and ratchet key [1].

Definition 14. (formal) Perfect forward secrecy means that an adversary compromising a long-term key in some epoch e^* cannot decrypt ciphertexts he obtained in epochs $e < e^*$ before that compromise.

In other words, forward secrecy protects the past session as depicted Figure 4.1. Even if in the future an adversary somehow gets a long-term key, he/she cannot affect the past session.

As an example consider the Diffie-Hellman since The Signal protocol uses it. It is known that the short-term keys are the base of Diffie-Hellman exponentials. If an attacker can access long-term key, the future communication will not be affected.

Post-Compromise Security. In forward secrecy, we thought about how attacks

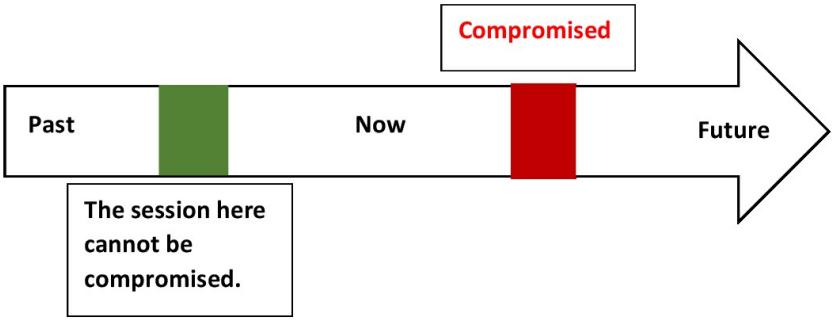

Figure 4.1: Forward Secrecy.

in the future will affect sessions in the past. Now we consider the scenario where Alice and Bob communicate and an adversary compromises their past session keys. In this case, is it possible for Alice and Bob to have a secure communication in the future (Figure 4.2)?

The ratchet mechanism of the Signal protocol has a fairly new technique called *"self-heal"*. The ratchet technique uses ephemeral Diffie-Hellman key exchanges for each session. The main purpose of the ratchet to be *"self-healing"* is to prevent an adversary to perform a successful attack. This process is also known as **"Post-Compromise Security"**: if an adversary gains access to an individual ephemeral key in the past or if an individual ephemeral key is considered weak, then the ratchet will *"self-heal"* which means any message sent after healing remains unintelligible to the attacker.

[10, 2] papers define post-compromise security:

Informally, a protocol between Alice and Bob provides **Post-Compromise Security** (PCS) if the communication between Alice and Bob has the security guarantee that even if the secrets that belong to one of them have been compromised, the communication is still unintelligible to an adversary after the *"self-healing"*.

Definition 15. (formal) An adversary compromising an individual ephemeral key in some epoch e^* cannot decrypt ciphertexts he obtained in epochs $e_{\text{after healing}} > e^*$ after that compromise.

The Signal protocol has the double ratchet mechanism which allows each communication key to be used to generate new keys and be deleted after use. This mechanism provides post-compromise security since if an adversary gets any of the ephemeral keys from a past session, it can not retrieve future information using that compromised key after the ratchet healed itself.

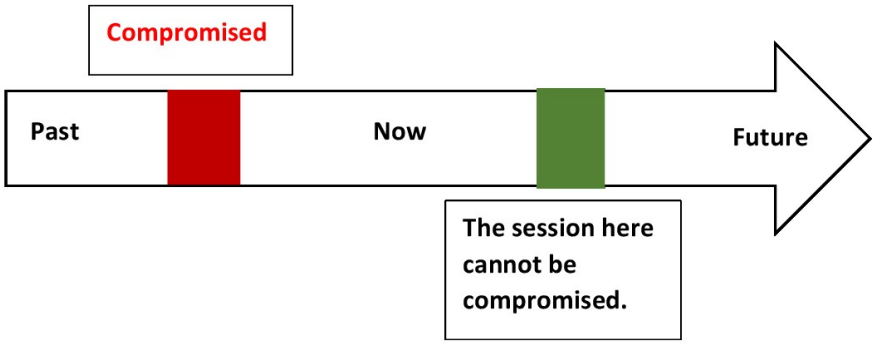

Figure 4.2: Post-Compromised Security.

4.2 Security Investigation of the Signal Protocol

The Signal protocol achieves some goals to guarantee the secure communication:

- Encrypted and authenticated messages;
- Forward secrecy.
- Post-Copromise security

To achieve these goals several techniques are used:

Signed and One-time prekeys. Assume both Alice and Bob are using just one-time prekeys. In this case, they will have a long-term identity key, i.e., a private and a related public key to identify the ID of Alice (or Bob) and a finite list of one-time prekeys. They will generate these keys and publish them in the key directory server. Then whenever Alice wants to send a message to Bob, she just gets one of those keys from the key directory, encrypts the message and sends it to Bob. Only Bob can decrypt this message since he knows both the long-term private key and one-time private prekey. After he decrypts the message, he deletes the one-time private prekey.

Even if this method helps to achieve forward secrecy there are some problems:

- Bob might run out of prekeys, since for instance he can generate 100 prekeys at a time. When all of them are used, he will generate more.
- The key directory server can lie to Alice about Bob's public prekey. Alice can obtain from the key directory server Bob's identity public key , but incorrect public prekey. In this case, Bob gets a message that he cannot decrypt. That broken message would also not achieve forward secrecy since an attacker that can compromise Bob's private key is able to decrypt the broken message.

Another technique can be that Alice and Bob sign their prekeys and define an expiration time after which the prekeys will no longer be valid. In other words: Bob signs his prekeys using his identity key pair and publishes them to the key directory. Alice can use those prekeys until Bob replaces them. Regularly, Bob updates his list of prekeys substituting an old prekey with a newly generated signed prekey.

The main problem is that Bob keeps the signed prekeys for a long period without deleting them.

To ensure secure communication the developers joined above mentioned methods. Eventually, to make the Signal protocol more secure, the parties should have a long-term identity key, a signed prekey and a fixed number of one-time prekeys.

When Alice wants to send a message to Bob, she gets his identity public key, public signed prekey and one one-time prekey. She uses all these keys during the key exchange phase.

Key Exchange protocol: is called "TripleDH" or "X3DH". The main purpose of this key exchange is to come-up with an agreed-upon shared secret key/master secret that is known only by Alice and Bob and has forward secrecy.

Alice and Bob publish to the KDS signed prekeys using their identity keys. They keep the old identity key until they publish new signed prekeys. After publishing new signed prekeys, Alice and Bob delete their old identity keys. This helps to guarantee the forward secrecy.

Double Ratchet: is the fairly new mechanism used by the Signal protocol to have forward secrecy and post-compromise security. After Alice and Bob calculate a shared secret; they use a double ratchet to secure messages. The main part of the double ratchet is KDF chain that is used by Alice and Bob to derive a root chain, a sending chain and a receiving chain. The KDF chain ensures forward secrecy by deleting the used keys and post-compromise security by performing "self-healing" which is the main property of the Signal ratchet.

AEAD: The Signal protocol uses AEAD encryption scheme to ensure the authentication and integrity of the message. AEAD encryption scheme prevents the communicating parties to use keys multiple times. If a key is used multiple times, the misuse-resistance is raised.

4.3 Weaknesses

Although the Signal protocol uses methods mentioned in Section 4.2 to ensure forward secrecy and post-compromised security, it has some vulnerabilities.

Man-in-the-Middle Attack: This attack involves an adversary that secretly relays and changes the communication between two parties that are not aware of the additional adversary. Assume Alice and Bob wants to communicate and Eve wants to know what they are talking about. The man-in-the-middle attack happens as follows (see Figure 4.3):

1. Alice sends a message "Hi, Bob. I am Alice. Give me your public key." and thinks she is talking to Bob.
2. Eve gets that message and sends it to Bob.
3. Bob reads that message and sends his public key (pk_B) to Eve. Bob also thinks that he is talking to Alice.
4. Eve sends his own public key (pk_E) to Alice instead of Bob's secret key.

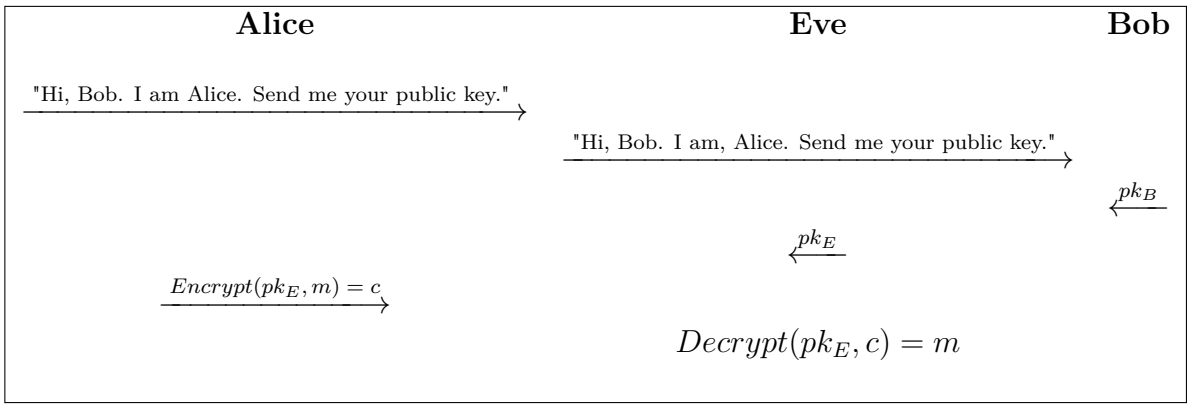

Figure 4.3: Man-in-the-Middle Attack.

5. Alice encrypts the message m using the Eve's public key and sends that message to Eve. Alice still thinks that she is talking to Bob.
6. Eve can decrypt and read the message since it has been encrypted with his public key.

The Signal protocol uses the KDS as an third trusted party. The parties that want to communicate need to send their public data to the KDS.

What if the KDS is corrupted and someone gets control of the KDS?

In this case, the man-in-the-middle attack can happen, i.e., the adversary that has control over the KDS can fake two conversations as shown in Figure 4.4. In this case, the malicious KDS becomes *"the man in the middle"*.

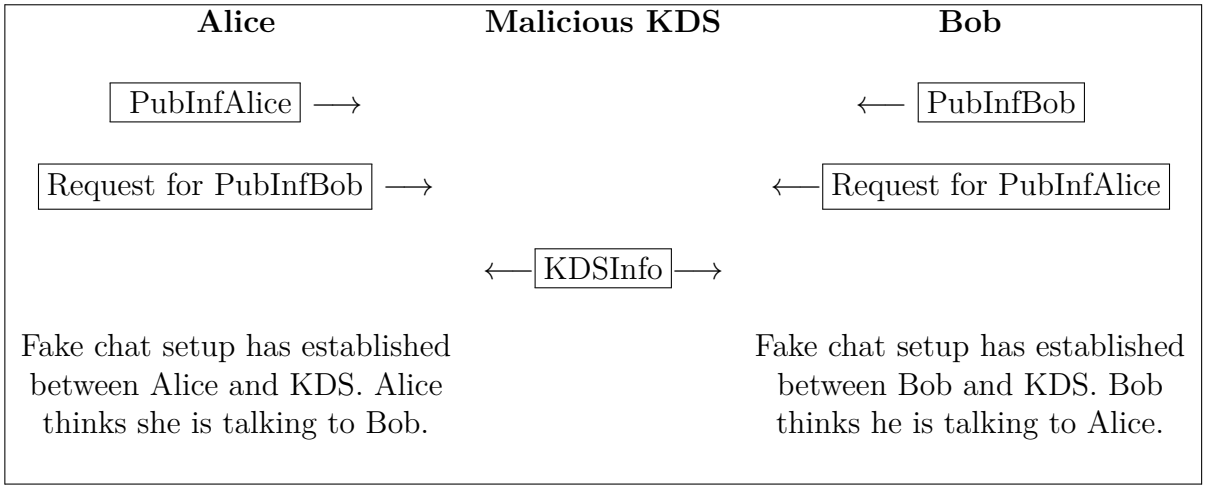

Figure 4.4: The malicious KDS performs the man-in-the-middle attack.

The development of the technologies increases the security problems of the data moving across the Internet and it is difficult to have an *"ideal"* mechanism to protect the data. However, the Signal protocol has above mentioned weak points, it is a great to tool that can still provide privacy. However, t
here are weaknesses with suggested mitigation methods that are discussed by the creator of the Signal protocol from Open Whisper Systems [59, 57]:

- The ratchet mechanism used in the Signal protocol provides forward secrecy security. In other words, this mechanism does not allow an adversary to track encrypted messages and use that data to compromise one of the communicating parties. The only way of compromising the communication between to users, requires the adversary to have access to the device and recover the deleted messages keys.
- The ratchet has a *"self-healing"* feature which protects the session if an attacker compromises one (or both) of the communicating parties and observes encrypted messages. However, a compromise of the secret identity key, stored in device, means lose of security of future communication. For instance, if the adversary compromises the device of one of the parties and obtain the user's identity key, the adversary can use it to create new communication session and pretend to be the compromised user. Therefore, it is necessary to be careful so that users should be careful, and if they observe compromission of their keys or devices, they need to derive keys or change the device.
- It is also possible that the keys that belong to skipped messages cause problems:
  1. A party can be forced to store a lot of skipped messages which allocate a large space of the memory. This can cause denial-of-service. In order to avoid this problem, the number of skipped messages per-session should be limited.
  2. There can be possibility of losing messages. An adversary might compromise a message from one user in order to get the skipped message keys which can cause the compromise of the session between the communicating parties. This problem can be solved by deleting the keys of the skipped messages using a timer or counter.
- During the asymmetric-ratchet phase, a ratchet key pair and a sending chain are derived. Since sometimes it is not necessary to use the sending chain immediately used after the asymmetric-ratchet phase is established, the asymmetric phase can be established when the sending chain key is needed. This shortens the lifetime of the ratchet keys which, in turn, increase their security. In other words, the derivation of the sending chain when it is not need might cause the lose of ratchet key if it remains for a long time until it is used.
- In order to authenticate themselves parties can either compare the safety numbers or scan QR codes. If for example, the safety numbers are not equal, this means the parties cannot authenticate themselves and the conversation is "interrupted" by another party. At the same time, it is impossible to get any information about the malicious party who they are communicating with.
- As it is explained in Chapter 3, when Alice sends her initial message to Bob that message should contain a one-time prekey. Otherwise, this message can be replayed to Bob several times. Since the messages have the same content, Bob will think that Alice sends him the same message several times. The initial message with a replayed one-time prekey, induce Bob to derive the same shared secret in different protocol runs. This can be solved if Bob either creates a blacklist of such messages or replaces the signed prekey rapidly or, the X3DH protocol have to use randomized encryption key before Bob sends

encrypted data.

- The mutual authentication and forward secrecy are guaranteed by the DH computations and by ignoring the prekey signature. This will cause *"weak forward secrecy"* attack: if the server is corrupted, one of the parties can get a forged prekey bundle. Then another party can also be compromised using his/her identity key and shared secret. The Signal protocol uses ephemeral keys and prekeys to ensure the security, if an adversary gets access to private keys, this will affect the security. If an attacker compromises the identity private key that belongs to one of the parties, he/she can impersonate that party. If an attacker compromises the prekey private keys that belongs to one of the parties, this will have effect on the security of the shared secret values.

5

Conclusion

Social networks have eased lives of millions of people by helping them to reach information faster and made them more connected. However, these services come with their cost in the form of threats to the security and privacy of social network users.

In the recent years, Snowden revelations and Facebook-Cambridge Analytica cases showed how social network users' private data is collected, stored, analyzed even without their consent. Especially, the Facebook-Cambridge Analytica data scandal has showed that by using this data, other parties not only identify people's demographic and geographic characteristics but also their psychographic tendencies to even target very important events such as 2016 USA presidential elections and Brexit referendum. These scandals have increased the awareness among people about how their data are exploited. It also proved that there should be more academic focus on privacy and security in order to make more progress on these domains.

The security of messaging apps that people use all around the world is a main concern these days. The components of these applications and their working principles can show us how secure they are. Therefore, it is important to have more awareness about specific parts of the applications such as messaging protocols. Considering the fact that the apps like WhatsApp and Facebook Messenger are the most popular messaging apps and they use the Signal protocol, it is interesting to investigate how it works and how secure it is.

This thesis work combined all the information that is sufficient to understand how the Signal protocol works. The cryptographic primitives, their properties and how they are deployed in the Signal protocol. Since the thesis is not only addressed to readers who have connection to cryptography and other areas in computer science, the information is provided in an easy way with examples. Then, the Signal protocol is explained with diagrams and an easier notation is proposed. The thesis work also combine and explain *"why"* the Signal protocol is secure. Going through the primitives that the Signal protocol uses, we explained the security properties that it achieves. For example, why it has forward secrecy and post compromise security and how it achieves them.

Future work. The topic of this thesis is fairly new and it can be extended further. Due to time limitation the scope of the thesis is kept as it is. So, we suggest several problems that can become the potential research topic in the future.

The other direction of this thesis can be the investigation of how the Signal

protocol works for group chat. Can it also achieve the same security properties for group chat? Although, it might still be secure even for group chat, the working principle of the Signal protocol differs for group chat. It is easy to understand how the Diffie-Hellman key exchange protocol works for two parties. If there are, for instance, 20 people messaging to each other, how does it calculate the shared secret among those people?

It might also be interesting, to investigate how the protocol is implemented in the different messaging apps. For example, analyzing the code or finding vulnerabilities in code. Algorithmic-wise, Signal is a secure protocol. However, when software developers use specific programming language to implement it, they may use vulnerable functions and methods. Therefore, it is necessary to have such kind of research and maybe suggest a secure functions or libraries to develop the Signal protocol for applications.

This thesis work has a brief information about the Signal app. It would be interesting, if all the features of Signal are investigated and presented to readers. OWS says that it is a secure app. But should we believe them? Why is it secure? Does using the Signal protocol makes it secure app? Are there other vulnerabilities that can help others to gain access to others data?

This thesis work has presented the Signal protocol with focus on explaining in a simple language how it works and how it achieves specific security properties proved by Cohon-Gorden et al's [11].

The development of technologies evolve continuously. Therefore, people should find new and more secure methods and build new protocols in order to protect their data across the Internet. Even though, it is a complex research topic, it has to encourage us to develop better and more secure communication protocols.

Bibliography

[1] S.A.Vanstone A.J. Menezes P.C. van Oorschot. "Key Establishment Protocols". In: *Handbook of Applied Cryptography*. CRC Press, 1997, p. 490. ISBN: 9781439821916.

[2] A.Lehmann and B.Tackmann. *Updatable Encryption with Post-Compromise Security*. Cryptology ePrint Archive, Report 2018/118. https://eprint.iacr.org/2018/118. 2018.

[3] *Android App Aims to Allow Wiretap Proof Cell Phone Calls*. URL https://www.forbes.com/sites/firewall/2010/05/25/android-app-aims-to-allow-wiretap-proof-cell-phone-calls/#3e6faecf7b1b, visited 2018-10-5.

[4] *Apple, Facebook, others defy authorities, increasingly notify users of secret data demands after Snowden revelations*. URL https://www.washingtonpost.com/business/technology/apple-facebook-others-defy-authorities-increasingly-notify-users-of-secret-data-demands-after-snowden-revelations/2014/05/01/b41539c6-cfd1-11e3-b812-0c92213941f4_story.html?noredirect=on&utm_term=.13ca2f88618d, visited 2018-08-10.

[5] B.I.A.Barry and F.M.Tom. "Instant Messaging: Standards, Protocols, Applications and Research Directions". In: Nova Science Publishers, 2009.

[6] B.Preneel. "Cryptographic Hash Functions". In: 1994.

[7] J.Boase B.Wellman A.Quan-Haase and W.Chen. "The Internet in Everyday Life". In: (2002). DOI: 10.1002/9780470774298.ch2.

[8] *Cambridge Analytica*. URL https://cambridgeanalytica.org/, visited 2018-09-02.

[9] H. Arshad Ch.Johansen A. Mujaj and J. Noll. *The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications (authors' version)*. Tech. rep. University of Oslo, 2018.

[10] K. Cohn-Gordon, C. Cremers, and L. Garratt. "On Post-compromise Security". In: *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. 2016, pp. 164–178. DOI: 10.1109/CSF.2016.19.

[11] K. Cohn-Gordon et al. "A Formal Security Analysis of the Signal Messaging Protocol". In: *2017 IEEE European Symposium on Security and Privacy (EuroS P)*. 2017, pp. 451–466. DOI: 10.1109/EuroSP.2017.27.

[12] *Digital Photography Review*. URL https://www.dpreview.com/forums?fbclid=IwAR3B8Rwv2c0amnNf9lQs1he4mME1N2AKbJBrGuLa7m0V19gWnOJITUTVXYE, visited 2018-08-10.

[13] *Edward Snowden: Leaks that exposed US spy programme*. URL https://www.bbc.com/news/world-us-canada-23123964, visited 2018-08-10.

[14] C.Forler F.Abed and S.Lucks. “General classification of the authenticated encryption schemes for the CAESAR competition”. In: *Computer Science Review* (2016), pp. 13–26.

[15] *Facebook Cambridge Analytica Scandal: 10 Questions Answered*. URL http://fortune.com/2018/04/10/facebook-cambridge-analytica-what-happened/?fbclid=IwAR0C2003k5Ot1gVBSMM3Xcm4FZ-8DBa8omejr8dSExjzjxEC7-GrwK8gG0A, visited 2018-09-01.

[16] *Facebook-Cambridge Analytica: A timeline of the data hijacking scandal*. URL https://www.cnbc.com/2018/04/10/facebook-cambridge-analytica-a-timeline-of-the-data-hijacking-scandal.html?fbclid=IwAR3p66Tkx3Pv2jk1A_SEfJJ6PZgGRMhqR6EXyF1QKcVTs34BPLXjnw67BA4, visited 2018-08-10.

[17] *Five years on, what has changed since the Edward Snowden scandal?* URL https://www.verdict.co.uk/snowden-scandal-five-years-gdpr/, visited 2018-08-10.

[18] Luke Garratt. *Realistic, Strong and Provable Key Exchange Security*. Tech. rep. University of Oxford, 2018.

[19] *Google Privacy and Terms*. URL https://policies.google.com/privacy?hl=en&gl=ZZ&fbclid=IwAR17YmILBn9KvXDz_j-Bss8vmBP-QJDRYc3hsPm\X86xQPQSsYA7SIcjS3DU, visited 2018-09-03.

[20] M.M. Grant and J.Cheon. “The Value of Using Synchronous Conferencing for Instruction and Students”. In: (2007).

[21] H.Krawczyk. “Cryptographic Extraction and Key Derivation: The HKDF Scheme”. In: *Advances in Cryptology – CRYPTO 2010*. Ed. by Tal.Rabin. Springer Berlin Heidelberg, 2010, pp. 631–648. ISBN: 978-3-642-14623-7.

[22] H.Krawczyk, M.Bellare, and R.Canetti. “HMAC: Keyed-Hashing for Message Authentication”. In: RFC Editor, 1997.

[23] H.Krawczyk, M.Bellare, and R.Canetti. “Message Authentication using Hash Functions”. In: 1996.

[24] *Internet Usage Statistics*. URL https://www.internetworldstats.com/stats.htm, visited 2018-04-15.

[25] J. Kim and J. W. Yoon. “Honey chatting: A novel instant messaging system robust to eavesdropping over communication”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 2184–2188. DOI: 10.1109/ICASSP.2016.7472064.

[26] Rubin Jan. *Security Analysis of the Signal Protocol*. Tech. rep. Department of Computer Systems, Czech Technical University in Prague, 2018.

[27] R. B. Jennings et al. “A study of Internet instant messaging and chat protocols”. In: *IEEE Network* (2006), pp. 16–21. ISSN: 0890-8044. DOI: 10.1109/MNET.2006.1668399.

[28] Y.Lindell J.Katz. In: *Introduction To Modern Cryptography*. CRC Press, 2015. ISBN: 978-1-4665-7026-9.

[29] K.S.McCurley. “The Discrete Logarithm Problem”. In: *IEEE Trans. Inf. Theor.* (1990).

[30] *Mark Zuckerberg's post on Facebook*. URL https://www.facebook.com/zuck/posts/10104972903079161?__tn__=-R, visited 2018-08-10.

[31] *Mark Zuckerberg's post on Facebook*. URL https://www.facebook.com/zuck/posts/10104899855107881?__tn__=-R, visited 2018-08-10.

[32] Nikos Mavrogiannopdulos. *Secure communications protocols and the protection of cryptographic keys*. Tech. rep. Department of Electrical Engineering, Katholieke Universiteit Leuven, 2013, p. 11.

[33] D.Wagner M.Bellare P.Rogaway. "A Conventional Authenticated-Encryption Mode". In: (2003).

[34] R.Canetti M.Bellare and H.Krawczyk. "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols (Extended Abstract)". In: In: 30th ACM STOC. ACM Press, 1998, pp. 419–428.

[35] *Messenger*. URL https://www.messenger.com/, visited 2018-04-18.

[36] *Messenger Starts Testing End-to-End Encryption with Secret Conversations*. URL https://newsroom.fb.com/news/2016/07/messenger-starts-testing-end-to-end-encryption-with-secret-conversations/, visited 2018-09-02.

[37] M.Marlinspike and T.Perrin. *The Double Ratchet Algorithm*. URL https://signal.org/docs/specifications/doubleratchet/, visited 2018-08-30. 2016.

[38] M.Marlinspike and T.Perrin. *The X3DH Key Agreement Protocol*. URL https://signal.org/docs/specifications/x3dh/, visited 2018-10-10. 2016.

[39] *Most popular global mobile messenger apps as of October 2018, based on number of monthly active users (in millions)*. URL https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, visited 2018-05-18.

[40] Aulon Mujaj. *A Comparison of Secure Messaging Protocols and Implementations*. Tech. rep. Department of Informatics Faculty of mathematics and natural sciences, University of Oslo, 2017.

[41] *NSA Bill Missing Key Reforms Passes House*. URL https://www.huffingtonpost.com/2014/05/22/nsa-bill-reforms-house_n_5372740.html, visited 2018-08-10.

[42] *Number of monthly active Facebook users worldwide as of 3rd quarter 2017*. URL https://investor.fb.com/investor-events/event-details/2018/Facebook-Q4-2017-Earnings/default.aspx, visited 2018-04-18.

[43] The European Parliament and the Council of the European Union. In: *REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. 2016.

[44] *Pizza Hut was hacked, company says*. URL https://www.miamiherald.com/news/nation-world/national/article178930896.html, visited 2018-10-23.

[45] P.Rogaway. "Authenticated-encryption with Associated-data". In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 98–107. ISBN: 1-58113-612-9. DOI: 10.1145/586110.586125.

[46] *Protecting democracy is an arms race. Here's how Facebook can help*. URL https://www.washingtonpost.com/opinions/mark-zuckerberg-protecting-

democracy-is-an-arms-race-heres-how-facebook-can-help-win-it/2018/09/04/53b3c8ee-b083-11e8-9a6a-565d92a3585d_story.html?utm_term=.2d408b883553, visited 2018-08-10.

[47] *Signal on the outside, Signal on the inside*. URL https://signal.org/blog/signal-inside-and-out/, visited 2018-05-05.

[48] *Signal Terms and Privacy Policy*. URL https://signal.org/legal/, visited 2018-11-02.

[49] *Skype*. URL https://www.skype.com/en/about/, visited 2018-04-18.

[50] D.Wind S.Schroder M.Huber and Ch.Rottermanner. "When Signal hits the Fan: On the Usability and Security of State-of-the-Art Secure Mobile Messaging". In: *1st European Workshop on Usable Security*. 2016. DOI: http://dx.doi.org/10.14722/eurousec.2016.23012.

[51] *Stack Overflow*. URL https://stackoverflow.com, visited 2018-08-10.

[52] T. Frosch and C. Mainka and C. Bader and F. Bergsma and J. Schwenk and T. Holz. "How Secure is TextSecure?" In: *2016 IEEE European Symposium on Security and Privacy (EuroS P)*. 2016, pp. 457–472. DOI: 10.1109/EuroSP.2016.41.

[53] *Tech companies step up encryption in wake of Snowden*. URL https://www.ft.com/content/3c1553a6-6429-11e4-bac8-00144feabdc0, visited 2018-08-10.

[54] *TechCrunch*. URL https://techcrunch.com/?fbclid=IwAR21OxSxDhZ69l-3_rprCKk1aq1dhYZzNURVKmLZF2KsCWzBhxGGEjkgVJQ, visited 2018-08-10.

[55] *Technical Information*. URL https://signal.org/docs/, visited 2018-05-10.

[56] *TextSecure Protocol: Present and Future*. URL https://www.youtube.com/watch?v=7WnwSovjYMs, visited 2018-04-23.

[57] *The Double Ratchet Algorithm*. URL https://signal.org/docs/specifications/doubleratchet/#security-considerations, visited 2018-11-10.

[58] *The Facebook and Cambridge Analytica scandal, explained with a simple diagram*. URL https://www.vox.com/policy-and-politics/2018/3/23/17151916/facebook-cambridge-analytica-trump-diagram, visited 2018-08-10.

[59] *The X3DH Key Agreement Protocol*. URL https://signal.org/docs/specifications/x3dh/#security-considerations, visited 2018-11-10.

[60] *This is everything Edward Snowden revealed in one year of unprecedented top-secret leaks*. URL https://www.businessinsider.com/snowden-leaks-timeline-2016-9?fbclid=IwAR32mQExYG8a0clCnQjki03LOyAjKltyWUopsQRdSwOUYAyKs--Yxu2msA0, visited 2018-09-01.

[61] *Timeline of Edward Snowden's revelations*. URL http://america.aljazeera.com/articles/multimedia/timeline-edward-snowden-revelations.html?fbclid=IwAR3_ESzgF2y1d2ew0WQPdPKDHyQaRrELpKjJHBgIZGt3gkWJJn6Rkv92K3I, visited 2018-08-10.

[62] T.V.Lokven. "Review and Comparison of Instant messaging Protocols". In: 2011, p. 5.

[63] N. Unger et al. "SoK: Secure Messaging". In: *2015 IEEE Symposium on Security and Privacy*. 2015, pp. 232–249. DOI: 10.1109/SP.2015.22.

[64] L. Brown W. Stallings. “Computer Security Concepts”. In: *Computer security : principles and practice*. Pearson, 2016, p. 13. ISBN: 978-0-13-377392-7.

[65] *What the government could actually do about Facebook*. URL https://www.vox.com/policy-and-politics/2018/4/10/17208322/facebook-mark-zuckerberg-congress-testimony-regulation, visited 2018-08-10.

[66] *WhatsApp*. URL https://www.whatsapp.com/features/, visited 2018-04-18.

[67] *WhatsApp Security*. URL https://www.whatsapp.com/security/, visited 2018-05-16.

[68] *Why you're getting so many emails about privacy policies*. URL https://www.vox.com/policy-and-politics/2018/4/5/17199754/what-is-gdpr-europe-data-privacy-facebook, visited 2018-08-10.

[69] *Wikitravel*. URL https://wikitravel.org/en/Main_Page?fbclid=IwAR37wo_D9x2Z9U6v_5_kdaq6ZntdBDSG1LlSa0ByI7-CvgO95XEH5be-u8s, visited 2018-08-10.

[70] Z. Xiao, L. Guo, and J. Tracey. “Understanding Instant Messaging Traffic Characteristics”. In: *27th International Conference on Distributed Computing Systems (ICDCS '07)*. 2007, pp. 51–51. DOI: 10.1109/ICDCS.2007.149.