Functional Specification

PROJECT NAME

|  |  |
| --- | --- |
| Student Name: | Liucija Paulina Adomaviciute |
| Student ID: | 21790411 |
| Student Name: | Eryk Zygmunt Styczynski |
| Student ID: | 21753851 |
| Date of Submission: | /11/2024 |

**Table of Contents**

[1. Introduction 3](#_Toc181728148)

[1.1. Overview 3](#_Toc181728149)

[1.2. Glossary 3](#_Toc181728150)

[2. General Description 3](#_Toc181728151)

[2.1. Product / System Functions 3](#_Toc181728152)

[2.2. User Characteristics and Objectives 3](#_Toc181728153)

[2.2.1. User Characteristics 3](#_Toc181728154)

[2.2.2. User Objectives 3](#_Toc181728155)

[2.3. Operational Scenarios 3](#_Toc181728156)

[2.4. Constraints 4](#_Toc181728157)

[3. Functional Requirements 4](#_Toc181728158)

[3.1. The X3DH Key Agreement Protocol 4](#_Toc181728159)

[3.2. The Double Ratchet Algorithm 4](#_Toc181728160)

[3.3. Placeholder 5](#_Toc181728161)

[4. System Architecture 6](#_Toc181728162)

[5. High-Level Design 6](#_Toc181728163)

[6. Preliminary Schedule 6](#_Toc181728164)

[7. Appendices 7](#_Toc181728165)

[7.1. Sources 7](#_Toc181728166)

# Introduction

## Overview

The main goal of this application is secure messaging. With Double Ratchet protocol combined with Extended Triple Diffie-Hellman key agreement protocol, this application offers secure, peer-to-peer messaging platform with clean and easy-to-use graphical user interface that does not rely on the security of third-parties, such as servers or the service provider. The goal is to not obscure the fact that the communication is happening between two users, but to secure the contents of the conversation.

## Glossary

**Chain keys** – the Key Derivation Function keys for receiving and sending chains.

# General Description

## Product / System Functions

1. Message encryption and identity verification using X3DH with The Double Ratchet Algorithm

2. File compression using Huffman Coding Algorithm

## User Characteristics and Objectives

### User Characteristics

This application is aimed at security and privacy-conscious users who have at least minimal computer skills. The user’s goal is to have secure and private conversations with other users without relying on servers or service provider. The typical customer will use other privacy-enhancing features such as secure browsing and limiting data sharing with third-parties, and carefully selects applications they use based on how much data they collect, share and who has access to that data. The main challenge for the user is to find reliable, trustworthy and secure open-source applications to use for daily activities such as conversations and file sharing. To achieve more privacy and security these users are willing to sacrifice some functionality and convenience.

### User Objectives

## Operational Scenarios

1. Log in

2. Identity verification between two actors

3. Sending message between two actors

4. Sending a file between two actors

5. Identity verification between multiple actors

6. Sending message between multiple actors

7. Sending a file between multiple actors

## Constraints

# Functional Requirements

## The X3DH Key Agreement Protocol

* **Description**
* **Criticality** – Essential
* **Technical issues**
* **Dependencies with other requirements**
* **Others as appropriate**

## The Double Ratchet Algorithm

* **Description**

The Double Ratchet algorithm is used by two parties (Alice and Bob) to exchange encrypted messages using a shared secret key. To prevent an attacker from stealing one party’s keys and decrypting future messages, the symmetric-key ratchet is combined with Diffie-Hellman ratchet which updates the chain keys based on the Diffie-Hellman output.

The parties derive new keys that have Diffie-Hellman calculation results mixed in for every message so that the later keys cannot be calculated using the earlier ones. If the key isn’t known, the output data is indistinguishable from random. Diffie-Hellman public values are sent attached to the message.

**Key Derivation Function**

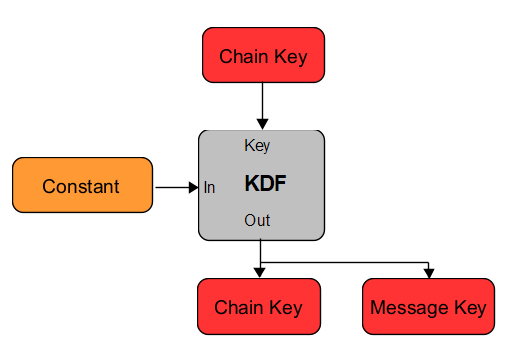
A Key Derivation Function is a cryptographic function that takes a secret and random KDF key and some input data and returns output data. In this application the secret KDF key is the output of the function defined in X3DH key agreement.

**Key Derivation Function Chain**

A Key Derivation Function chain is when some of the output from KDF is used as an output key and is used to replace the KDF key, which then may be used with another input.

**Symmetric-Key Ratchet**

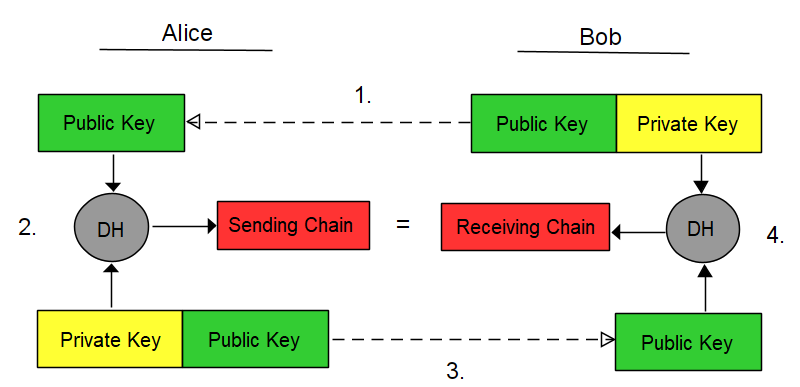
Every message is encrypted with a unique message key. The message keys are output keys from the sending and receiving KDF chains. The sending and receiving chains ensure that each message is encrypted with a unique key that can be deleted after encryption or decryption. Calculating the next chain key and message key from a given chain key is a single ratchet step in the symmetric-key ratchet.



Single step in symmetric-key ratchet

**Diffie-Hellman Ratchet**

As Alice and Bob exchange messages, they also exchange new Diffie-Hellman public keys and the output secrets of the Diffie-Hellman become the inputs to the root chain. The outputs generated during each Diffie-Hellman ratchet step are used to derive new sending and receiving keys.

****

Single Diffie-Hellman ratchet between Alice and Bob

During the Diffie-Hellman ratchet Alice and Bob, the ratchet happens in four steps:

1. Alice initializes using Bob’s public key.

2. Alice performs Diffie-Hellman calculations using her private key and Bob’s public key, which results in sending chain key.

3. Alice sends a message to Bob and advertises her public key to him.

4. Bob performs Diffie-Hellman calculations using his private key and Alice’s public key extracted from the message. This results in receiving chain key equivalent to Alice’s sending chain key.

After this, Bob repeats the ratchet using Alice’s advertised public key to calculate new sending chain key using a new pair of his public and private keys

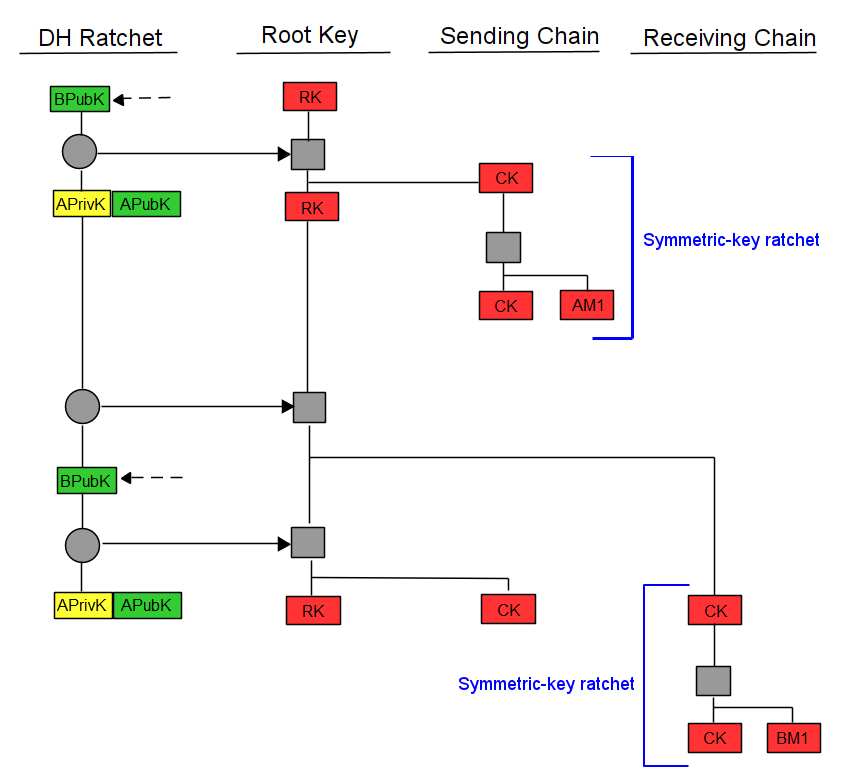
In a Double Ratchet session between two users, each party stores a Key Derivation Function key for three chains: a root, a sending and a receiving chain. One party’s sending chain matches other’s receiving chain.

**Double Ratchet**

The Double Ratchet is a combination of symmetric-key ratchet and Diffie-Hellman ratchet.

When a message is sent or received, a symmetric-key ratchet step is applied to the sending or receiving chain to derive the message key.

When a new ratchet public key is received, a Diffie-Hellman ratchet is performed prior to the symmetric-key ratchet to replace the chain keys.



Single Double Ratchet between Alice and Bob

Abbreviations:

* RK – Root Key.
* CK – Chain Key.
* BPubK – Bob’s Public Key.
* APrivK – Alice’s Private Key.
* APubK – Alice’s Public Key.
* BM1 – Bob’s First Message’s Key.
* AM1 – Alice’s First Message’s Key.

In Double Ratchet between Alice and Bob, Alice initializes with Bob’s public key and the initial root key (shared secret). As part of the initialization, Alice generates a new ratchet key pair using X3DH and feeds the output to the root KDF to calculate a new root key (RK) and sending chain key (CK). The old root key can be deleted. When Alice sends her first message, she applies a symmetric-key ratchet to her sending chain key. The output is a new message key (AM1).

When Alice receives a message from Bob, it will contain a new ratchet public key. Alice applies a Diffie-Hellman ratchet step to derive a new receiving and sending chain keys. Then she applies a symmetric-key ratchet step to the receiving chain to get the message key (BM1) for the received message.

* **Criticality –** Essential
* **Technical issues**

No Java libraries available for the algorithm – it will need to be implemented from scratch.

* **Dependencies with other requirements**

The Double Ratchet algorithm depends on the output of X3DH.

## Placeholder

* **Description**
* **Criticality** – Essential
* **Technical issues**
* **Dependencies with other requirements**
* **Others as appropriate**

# System Architecture

# High-Level Design

# Preliminary Schedule

# Appendices

## Sources

Marlinspike, Moxie. “The X3DH Key Agreement Protocol.” Open Whisper Systems, 4 Nov. 2016, https://signal.org/docs/specifications/x3dh/. Accessed 5 Nov. 2024.

Marlinspike, Moxie. “The Double Ratchet Algorithm.” Open Whisper Systems, 20 Nov. 2016, https://signal.org/docs/specifications/doubleratchet/. Accessed 5 Nov. 2024.