Signal Protocol Implementation using C#

Gaurav Palande (014861923)
Ashley Chen (014867175)
YC & GAP Security Services Pvt. Ltd.

12 september 2016

1 Motivation and Problem Statement:

Signal has a great user interface and has proven to make private communication simple by implementing a full end-to-end encryption for users of its service. The open source protocol library in Java is available at `https://github.com/whispersystems/libsignal-protocol-java/`.

However, the equivalent official C# implementation of the library is not currently available.

2 Proposed Solution/Approach:

The purpose of this project is to create the C# implementation of the library and evaluate as a team, factors such as usability and security in message communication under the Signal protocol. In particular we will seek to understand the various features that makes Signal so usable and secure.

We hope that by diving into the implementation, we can understand a bit more about how the protocol works. We are bound to arrive at a suitable conclusion and a possible step forward for protocol through this implementation. If time permits, we also plan to look into the Noise Protocol Framework which has exhibited numerous advantages at the transport layer for the Signal protocol.

3 Details of implementation:

- **Platform:**
  The chat application will be a web application.
- **Programming Language:**
  The application will include the Signal protocol *C# implemented library*. The application itself will be developed using *.Net framework* and *Visual C#*
- **Different components involved in your project:**
  The project involves the use of signal protocol implemented using a **C# library**. There is a possibility of this library being used as a **web service**. The application will be simple message transfer with **limited size text** messaging functionality and possible

group chat functionality. The application will contain a **server side implementation** that assists in distributing the messages among the **client applications**. The use of **encryption keys** as specified in the signal protocol such as,

  - **Pubic Key Types:** Identity Key Pair, Pre Key and One-Time Pre Key.
  - **Session Key Types:** Root Key, Chain Key and Message Key.

  The implementation of the key structure and encryption mechanism such as *Cure25519*, *Elliptical Curve Diffie-Hellman (ECDH)*, *AES-256* and *HMAC-SHA 256* will be integrated as part of the overall encryption mechanism.

- **Hardware requirements:**
  Basic hardware for running laptop/PC Web browser like Google Chorme .i.e.

  - **Windows:** Minimum Core i3 or equivalent running Windows Vista or later.
  - **Apple:** Macbook Pro 2010, Macbook Air 2011 running OS X 10.9 or above.

4 Detailed Time-Line

The project will follow a 4-Phase implementation process as mentioned by the professor.

- **Phase 1:** Project requirements and design documentation
- **Phase 2:** Server-side implementation
- **Phase 3:** Client-side implementation
- **Phase 4:** Encryption mechanism implementation

5 Description of Workload Distribution

- **Ashley:**
  - Generation of **Public Key Types**: *Identity Key Pair*, *Signed Pre Key and* and *One-time Pre Key*
  - Client registration functionality
  - Functionality needed for setting session and receiver
  - Calculating *Message Key* from *Chain Key*
  - **Optional functionality:** Key verification using QR code
  - Implementing transport level functionality using Noise Protocol framework
- **Gaurav:**
  - Generation of **Session Key Types**: *Root Key*, *Chain Key* and *Message Key*
  - Session initialization functionality
  - Message transfer functionality
  - Calculating *Chain Key* from *Root Key*
  - **Optional functionality:** Group chat functionality and key verification using fingerprinting (60 digit number)