650.445/600.454: Practical Cryptographic Systems October 4, 2016

Assignment 2

Instructor: Matthew Green *Due: 11:59pm, October 13*

Name: _____

The assignment should be completed individually. You are permitted to use the Internet and any printed references.

Please submit the completed assignment via Blackboard.

Problem 1: Implementing a Messaging Client (100 points).

You have been asked to implement a client for the JMessage encrypted instant messaging service. JMessage is a client-server architecture that allows individual clients to communicate using end-to-end encryption. The detailed operation of JMessage is described further below. To assist you in your implementation, we have provided you with three components:

1. A complete specification for the JMessage system
2. A working JMessage server, provided as a Docker image
3. A Java-based JMessage reference client that you can use to test your own implementation

Note that your implementation must be cross-compatible with the reference server and client. *In the (un)likely event that the specification proves inconsistent with the reference client/server, you should prioritize compatibility with the reference implementation.*

You may find the server and reference client at the following locations. You will need to use Git and install a Docker client on your machine,[1] along with some standard tools. Both resources include instructions for building and running the components:

1. Server: `https://github.com/matthewdgreen/jmessage-server` (clone the Git repository and follow the instructions in README.md)

2. Reference client: `https://github.com/matthewdgreen/jmessage-client` (this contains a Jar file you can run on the command line using the JRE)

Submission and grading: As a deliverable, you must provide the source code for your client. Your code must compile on one of the standard graduate lab machines (MSSI, UGrad, or Masters lab). If you are using a language that is not present on these machines, and you feel this is absolutely necessary, you may submit a VM image that contains all necessary tools. You may include a single script entitled **build** (*e.g.,* **build.sh**) that performs all actions necessary to compile your software. You must also include a simple README that

[1] https://www.docker.com

explains how to run and use your code. *If your code does not meet these requirements you will get a* 0 *on this assignment.*

Note: Particularly nice submissions will be considered for extra credit. This includes things like mobile clients, GUIs and additional features. Irrespective of these additions, you client must support the core message transmission functionality.