AnonTalk - an Anonymous Communication Application Powered by Dissent

Danqing Liu
Yale University CPSC 490, Spring 2013
Advisor: Bryan Ford

1. Introduction

AnonTalk is an anonymous communication application on the iOS platform powered by the Dissent system, which is an accountable anonymous group communication system under development at Yale. Dissent builds on dining cryptographers and verifiable shuffle algorithms and offers provable anonymity guarantees, while also seeks to offer accountable anonymity, which means that groups are protected from anonymous abuses. As Dissent achieves such strong goals, it is inevitably a complex system. And as of today, it still stays mostly as a research project rather than a production library.

AnonTalk is one of the first practical applications of Dissent. Thus the purpose of this project is twofold. First, it serves as a practicality proof that the Dissent technology can be ported to a mobile platform. As smart phones and mobile computing are becoming the main way of communication, it is essential to show that such a complex system can actually be run on mobile devices, which are much weaker in processing power and have much more limitations (in the case of iOS, partly due to Apple's restrictions). The mobile platforms are also based on ARM, which is entirely different from the desktop x86/64 architecture, so the portability of Dissent is also a critical concern. Second, it serves as a demo of the major use case of Dissent: as an anonymous chat application. If Dissent were to become a production system, it is likely to be used like what AnonTalk implements. The modular architecture of AnonTalk therefore provides a way for future developers to quickly include Dissent as part of their own projects.

2. Implementation

2.1. Overview

AnonTalk is a combination of Dissent, a system written primarily in C++ that serves as the server, and the iOS app itself, written in Objective-C/Objective-C++ native to iOS’s Cocoa Touch framework, that provides the user interface. As the two components are largely independent of each other, AnonTalk is designed in a modular fashion. In other words, both parts can be modified separately; the updated versions can then be recombined and the application will still work. The two parts of the app are connected through a simple interface contained in the Application.hpp file:
Dissent parameters can be passed in just as in the Terminal, and the server will be started (in a separate thread) at the desired port. The front-end of the application can then talk to the server through standard HTTP protocols. The front-end can also connect to a remote server if the user knows its IP address. The details of the AnonTalk implementation are discussed below.

2.2. Back-end structures

Most backend features are supported through a single Objective-C++ class: ATGlobalHelper. This is to conform to Apple's recommended Model-View-Controller scheme for iOS applications. There are three major methods in this class:

+ (void)submitMessage:(NSString *)message forController:(UIViewController *)controller usingBlock:(void_block_t)block;

This method submits the message the user entered to the specified Dissent server, via HTTP POST. The server address and port number are stored in permanent storage and thus do not need to be passed in. The last parameter, a void block, supports callback functions to be executed when the method returns.

+ (void)getMessagesWithOffset:(NSUInteger)offset andCount:(NSUInteger)count forController:(ATChatViewController *)controller usingBlock:(dictionary_block_t)block;

This method pulls count messages starting from the offset for the specified controller, and returns the result as a dictionary (key-value pair) in the callback block, which is the last parameter. The offset is calculated by counting the number of messages AnonTalk has in its database.

The above two methods are powered by AFNetworking, a networking library designed for iOS.

+ (void)startServer;

This is probably the most important method in the entire application. It connects the two pieces, the chat app and the Dissent server, together by calling the dissent_main function in this Objective-C++ class. This method utilizes a technology called Grand Central Dispatch (GCD), which provides thread support and concurrency control. When the user chooses local server, the Dissent server will be started in the background by GCD, and it will listen to incoming requests until it is shut down. As this is the only link between Dissent and Chat, the two components can be separated into two modules in a neat way.

The other component of the backend is the database that stores messages. This is achieved via the Core Data framework of iOS. To achieve consistency, only one instance of the database is kept and can be
accessed globally. The app currently only accesses the database from one single point. However, when
the app potentially grows in the future, having multiple handles to the database will be very dangerous
and error-prone. The proper way to access the database is set up in the ATGlobalHelper class.

The Message table contains three columns:

- **data**: the actual content of the message.
- **time**: the time the message is received. As we will discuss below, it is a design decision that we do not
  store outgoing messages when they are sent. As Dissent currently does not pass send time, AnonTalk
  uses receive time to order the messages.
- **conversation**: the conversation the message belongs to. This is currently set to the IP address of the
  server, since when the server IP changes, the messages will also disappear.

To facilitate future schema changes, database-related operations are implemented via an Objective-C
category that extends the Message class (the underlying class of the Message table). In this way, as long
as the above three attributes are kept, minimum amount of code needs to be changed when there are
any changes to the schema of the database.

There are four ways to alter the Message table:

+ `(Message *)messageWithContent:(NSString *)content conversation:(NSString *)conversation andTime:(NSString *)time inManagedObjectContext:(NSManagedObjectContext *)context;

This method adds the received message into the table. The

+ `(void)removeAllMessagesInManagedObjectContext:(NSManagedObjectContext *)context;

This method removes all messages in the database. It is provided for future use. The application does
not call this method anywhere currently.

+ `(void)removeAllMessagesForConversation:(NSString *)conversation inManagedObjectContext:(NSManagedObjectContext *)context;

This method removes all messages that belong to a certain conversation, which is currently defined by
IP address of the server. Suppose we stopped and relaunched the local server, we will need to purge the
old messages.

+ `(void)removeMessage:(Message *)message inManagedObjectContext:(NSManagedObjectContext *)context;
This method removes the specified message in the database. It is provided for future use. The application does not call this method anywhere currently.

2.3. Front-end structures

Overview of the layout of the application

The front-end of AnonTalk contains the following views:

- List of chatrooms (ATViewController)
  
  This view is the homepage of the application. Unlike the web interface, AnonTalk supports multiple chatrooms, distinguished by their IP addresses. For demo purposes, the app allows up to 10 different chatrooms, 1 being the local chatroom. Both numbers can easily be increased if necessary. However, as the Dissent encryption algorithm is very CPU-intensive, it is not a good idea to run more than 1 local chatrooms concurrently.

- List of messages in a specified chatroom (ATChatViewController)
  
  When the user taps on a chatroom in the table provided by ATViewController, the app segues to a chat interface provided by ATChatViewController. At the same time, a network call is
turned on in the background to retrieve new messages posted to the chatroom. This network connection is always on: as long as the user stays on the chat page, AnonTalk will prompt the server for new messages. Timeout of network connections has been turned off for this matter. This connection will only be closed when the user leaves the page, by clicking on the back button. For performance and usability reasons, only the currently active chatroom receives incoming messages.

It is worth pointing out that to guarantee anonymity, only incoming messages are stored. In other words, AnonTalk does not store outgoing messages; it waits until the server returns such messages back and then stores them just as the messages from other clients. Also for the same reason, Dissent only displays the receiving time of the messages, but not whether they were sent by the user or some other party. If the user wants otherwise, it can be easily altered to achieve such behavior. The database can contain a column for this purpose, although it is not included for now.

• Settings page (ATSettingsViewController)

AnonTalk provides a settings page in which the user can determine how to connect to the Dissent server. The user first chooses whether a local Dissent server should be used (defaulting to address \texttt{127.0.0.1:8080}), or a remote Dissent server should be connected. As the Dissent server is resource-intensive, the local server is only started when the user chooses to use it. On the other hand, remote servers provide a weaker anonymity guarantee. If the user chooses to connect to the local server, the settings of the server can be edited directly from this page. If the user chooses to connect to a remote server, the IP address and port number of the server can then be specified. These values are stored as \texttt{NSUserDefaults} and will be remembered across different launches of the app.

• Login page (ATLoginViewController) - not linked

A login page is also included for future use. Dissent does not yet support logging in, and for obvious security reasons (Dissent is an anonymous communication scheme), it may not support logging in in the future. In case such functionality is desired, the login page can be presented
modally before the list of chatrooms is shown to the user. Then the chatrooms available to the user will be automatically shown and connected.

2.4. Miscellaneous

AnonTalk is designed to be a ready-to-ship application. Therefore, it further contains:

- Retina-ready graphics and UI elements that work on iPhone 4 and later.
- Auto-layout support so it works full-screen on iPhone 5 as well as earlier generations of iPhone.
- App icon and launching image ready for App Store.
- A network activity indicator class for usability considerations.

3. Discussions and Future Improvements

While the server can be run on the iOS device, and while efforts are made to include only the essential components of the communication package, the speed of the local server is still unsatisfactory. It takes about 10 seconds for the server to be truly started and accepting requests, and each request has about 2 seconds delay. As smartphones becomes faster, this is gradually becoming a lesser issue, but we should still investigate ways to improve the performance.

One major weakness of AnonTalk is that once the local server cannot be stopped unless the user quit the application. This is due to Apple’s limitation on GCD and NSOperation. Even though the two technologies support canceling their respective threads, the underlying operation, in this case the Dissent server, needs to implement a cancellation method. Aborting a server on a desktop works just fine, but force killing a thread on iPhone has dangerous consequences. Therefore, in the future a method to stop the server is desirable.

Appendix - Compilation Instructions

(As Dissent is still under active development, this section is provided to facilitate the work of future developers of the system.)

AnonTalk is compiled for the iOS 6 platform and uses a Dissent version that depends on the following two libraries:

- Qt 4.8.4 - iOS build with SSL
• Cryptopp 5.6.2 (also known as Crypto++)

If you make changes to the Dissent system, you will need to recompile Dissent to generate the
libdissent.a static library to include in the AnonTalk project. To do so, you need a Mac running the
latest XCode with Command Line Tools package installed. You are strongly recommended to use the Qt
and Cryptopp packages that come with AnonTalk, because some special tweaks to the libraries were
performed to make sure they compile with iOS. You will need to:

1. (Only need to perform once) Install Qt SDK to the /Developer directory. As the configuration files
   are pointing to absolute paths, you have to install to that directory in order for Qt to work. To do so,
simply do:
   
   sudo tar -xf Qt4iOS-SDK-GLES2-SSL-#####.tar.gz -C /

2. You will need to compile Dissent using Apple's LLVM-GCC 4.2 compiler. To tell Qt to use the right
   compiler, type in the following commands:
   
   cd /Developer/Qt4iOS/qt-everywhere-ios-gles2-4.8.4/mkspecs

   sudo rm default

   sudo ln -s unsupported/macx-iosdevice-llvm default

3. Now you can compile Dissent (in its root directory) by typing
   
   /Developer/Qt4iOS/qt-everywhere-ios-gles2-ssl-4.8.4/bin/qmake application.pro

4. The above command will generate the makefile for you. In the auto-generated makefile, -Werror flags
   need to be taken out for the Cryptopp library contains some warnings.

5. When make succeeds, you will see the libdissent.a file in the root directory of Dissent. Open
   AnonTalk project in XCode and replace the old libdissent.a file. Now compile AnonTalk in XCode
   and you will be using the updated version of Dissent.

Note: If you are modifying the original version of Dissent, you need to rename the main function in
Application.cpp as dissent_main, and #include “Application.hpp”. The header file is the entry
point through which AnonTalk accesses Dissent, and it contains the following declaration:

   int dissent_main(int argc, char **argv);

The necessary Qt and Cryptopp libraries of the versions specified above have been compiled and linked
to AnonTalk. If you use the same versions, you do not need to reconfigure these linkers. However, if you
use other versions, you need to make sure that they are linked to AnonTalk.

Note 2: Apple XCode has 3 different types of compilers: GCC, LLVM-GCC, and clang. All the original
packages (Qt, Cryptopp and Dissent) are built using GCC, but Apple is dropping the support of GCC.
GCC is by default not included in the current version of XCode (4.6.2). Therefore, in order to compile properly, you will need:

- LLVM-GCC when generating the `libdissent.a` static library. Dissent uses some GNU extensions, so clang will not compile Dissent correctly.

- clang for AnonTalk. In XCode build settings, clang and its native C++ std library are necessary for AnonTalk to compile correctly. The build instruction file included has the proper settings. In particular, only Cryptopp 5.6.2 and above fully supports clang, and so earlier versions will not work properly.