Chapter 6

Conclusion

This chapter draws some conclusions from evaluation results achieved from user testing session and summarizes some criticisms of the project. The future work planned for the Secure Dropbox is then proposed.

6.1 Results

6.1.1 Was An Effective User End Encryption Tool Developed?

The best approach to appropriately illustrate the idea of user end encryption tool and asymmetric cryptography based secure sharing mechanism is to build a prototype which was well designed and implemented. Secure Dropbox accomplishes such a prototype and the most key idea in accordance with security has been expressed. Secure Dropbox does not invent anything but applied a cryptography combination in a new application scenario which has never been done. Its security is based on classic, theoretically strict, widely used and universally accepted cryptography algorithm. The design makes it zero-knowledge software: Secure Dropbox would never have access to the file or even the encryption keys. All static stored data that should be kept as secrets are unquestionably stored confidentially and when these data are decrypted, Secure Dropbox will not try to access anything. It undeniably gains users’ confidence when using public cloud storage to store their confidential documentations according to the user testing result. Secure Dropbox users would never work with unencrypted files in their Dropbox and these files are ready to share without much effort but only a small conversion upon the encryption key. Secure Dropbox could already be thought as a successful user end encryption tool to some extent.

However, it is still far from being a commercial software product. It provides higher level of security but at the same time disables some important features of Dropbox like version control and file recovering. Actually, to most uncommercial users, the stability and fault-tolerance of a cloud storage are often considered as much as its security. The trade-off will be absolutely there until these problems have been solved. For now everything of Secure Dropbox is only in prototype level.

6.2 Criticism

6.2.1 Design Criticism

Although it expresses the main idea of Secure Dropbox, the project is designed only for demonstration purpose. It lacks some essential features that could actually make better idea expression. For example, a logging module is basically built in any software product. The potential fault-tolerance and error recovering features of Secure Dropbox could have been illustrated by implementing a logging module inside. It also lacks of consideration about availability like it was designed to grant user with a minimum file reading permission under Secure Dropbox local mode. Although the Dropbox client based synchronization mechanism works robustly, no fault recovering or feedback happens when error occurs since Dropbox client is not designed to be based on by other application and there is no programmable interface provided by the Dropbox client application. Dropbox core API is the only recommended way to implement a third-party application of Dropbox.

6.2.2 Implementation Criticism

The user space cryptography implementation essentially narrows down the file that could have been supported by Secure Dropbox. For this stage only text file is supported and supporting new types of file requires considerable efforts. A file system level encryption implementation would solve this problem in essence. Additionally, a more flexible configuration interface should be made like allowing user to configure their own cryptography application schema based on different environment and practical requirements. User interface should be improved by not only redesigning based on some human computer interaction principles but also with advanced user interface design elements integrated.

6.2.3 Testing Criticism

Performance testing lacks of comparison with other algorithm or under different application environment so that these numbers are not so meaningful and does not speak much about performance of Secure Dropbox. In addition, the user experience testing lacks of expert participants although some of them have computer science background. Security expert usually have better understanding in this area and able to propose constructive expertise about such a software product.

6.3 Future Work

6.3.1 Version Control and File Recovering

Version control and file recovering service provided by Dropbox is disabled because there is no corresponding file encryption key version control mechanism in Secure Dropbox. A possible design could be padding the file name with timestamp and using this file name as key of version control table. This table records the history of certain file and its corresponding encryption key.

6.3.2 User Profile Management

In terms of user profile management, only user registration and login have been implemented. The following job will be implementation of a more generic profile management module with regular features like logout, account cancellation and more importantly password modification. The RSA key pair is a part of user profile as well. The updating mechanism towards expired RSA key pair will be implemented as well.

6.3.3 More Secure File Sharing Mechanism

File will keep using the same key until its source text has been modified and reloaded via Secure Dropbox again. For now file sharing does not change the file encryption key not only the sharing url could be cancelled or expires after certain time slot but also could be accessed only by performing valid authentication and through some designed user interface control. However, a new file sharing mechanism provides one time encryption key could guarantee a better security for the file whose file encryption key has been known to other users.

6.3.4 File Sharing Refreshing

File sharing url generated by Dropbox expires in 3 hours. This is because the OAuth token expires that time as well so Dropbox does not want third-party application holding an expired access token could still access the shared file. To keep the file sharing until further cancellation, the file sharing record in the data base could be refreshed periodically to update the previous url with the newly generated one. However, calling /media interface requires a valid token which have to be fetched by manual authentication, the mechanism of keeping the access token valid calls for more investigation.

6.3.5 File System Level Encryption

Secure Dropbox currently supports operation upon text file only because the user space encryption makes adding new supported file type needs lots of effort especially when some special file format involves. File system level encryption makes cryptography procedure transparent to user space applications. Dragging files into the specific folder with customized file system level system calls would perform encryption automatically and vice versa.

6.3.6 Configuration Interface

Now Secure Dropbox configuration could be carried out by changing parameters in certain Python source code. However, an embedded configuration interface inside Secure Dropbox client would guarantee a better availability and robustness since parameters might be modified into some illegal expressions or invalid values which causes availability problem. Configuration interface could limit the options for configuration and execute parameter checking before new configuration taking effect.

6.3.7 Multi-platform Implementation

There are lots of Dropbox users who want to synchronize their files between different terminals include those portable terminals like smart phone or pad computer. Since Secure Dropbox is implemented with Python, it would not be extremely time consuming to do the application transplantation between different operating system. Python is also supported in some portable operating systems like Android and IOS. The difference to be concerned about is the computation and network capacity which would impact the performance a lot.

6.3.8 Local KMS Mechanism

Now Secure Dropbox local mode works based on KMS files stored in local file system which is generated after last login. It provides key chain and RSA key pair required in order to execute local reading operation. Nevertheless, a better designed local KMS mechanism should guarantee Secure Dropbox users exactly the same performance as regular mode does. An optimized local KMS mechanism is designed to perform delayed KMS information updating when Internet access revives.