FaceDrop: ensuring secure sharing in the cloud

Ivan Cherapau
Department of Electrical and Computer Engineering
University of British Columbia
icherapau@ece.ubc.ca

Primal Wijesekera
Department of Electrical and Computer Engineering
University of British Columbia
primal@ece.ubc.ca

Konstantin Beznosov
Department of Electrical and Computer Engineering
University of British Columbia
beznosov@ece.ubc.ca

ABSTRACT

Cloud storage has become the most popular way for data backups and sharing information. With the increased usage of cloud storage, number of storage compromises skyrocketed as well leaving a huge number of business and personal information at risk. At the same time placing trust on cloud service providers has been a centre of discussion lately due to various reasons. To this end, we present FaceDrop a solution that reduces the necessity of the amount of trust needs to be placed on the cloud storage provider. FaceDrop allows people to securely share data among trusted parties and also to backup the data using untrusted service providers. This also gives the feature to consolidate available cloud storage space. The goal of minimum trust is achieved by using as many channels as possible while sharing data from authentication to file transferring. FaceDrop is implemented using popular social networks and cloud service providers. It gives a more practical threat model where cloud service providers only needs the minimum trust from the end users.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Miscellaneous; E.3 [**Data**]: Data encryption

1. INTRODUCTION

Cloud Computing revolutionized the computing world. With the emergence of Cloud Computing, cloud storage is one of few domains that has attracted many customers. People use cloud storage for many different reasons such as to back up data, share data among people, etc and many providers offer a wide variety of flexible online data storage services. These services have many advantages - the data can be accessed from any location connected to the Internet, it can be synchronised across multiple devices and easy to share.

With the increase usage, the amount of trust people used to place on providers has been deteriorated due to increase number of compromised and with political reasons such geographical lows that are to be met. Although the advantages of using cloud is unquestionable, there are many things that can go wrong. When users use clouds, they release control over their data, because IT infrastructure belongs to the cloud provider. Cloud service providers can have some downtime, they can be out of business, data can be modified without user's notice due to the provider malfunction or an attack. Cloud providers have excessive control over users data. However cloud storage is growing rapidly and has been like that for sometime.

With this growing market, quite number of third party services started to emerge in the market giving the notion of a secured cloud storage. Among many solutions email providers come first with their increasing number of space given for attachments. Free email providers such as GMail, Yahoo and Hotmail attracts many people due its usability. ThirdParty web based services such as SendSpace, Filemall let people to use a web site interface to upload and download data when sharing. Peer to Peer (P2P) systems offers great deal of support in sending bulk of data but they do not use cloud at all and sometimes with these approaches we do not have a control over where data is being host. The last and most popular service is third party application exploiting the cloud storage and also giving the notion of a shared folder in the local disk. Service like DropBox, SkyDrive have attracted millions of users with thier cloud storage offerings.

While each of these service provide an unique set of features, they also lack in some sort of a security feature that is becoming increasingly needed. Most of the providers they do not give the ability to encrypt the data by using custom key un-known to the provider specially in the event of data sharing. And while users tend to have many cloud service accounts, tools to consolidate those available space securely are rare to find. With the increase threat of both insider and outside intrusion leaving an entire data chunk at one place is a huge security vulnerability that can pose a great threat. All these reasons lead to a low level of trust on keeping and sharing data on the cloud, comparing with conventional offline storage, where users have control on the infrastructure.

To this end, we propose FaceBox. It's goal is to create new approach that allows users to have secure storage and sharing over untrusted cloud providers. The general idea of the proposal is to use separate channels for the authentication and for data exchange. Authentication and key establishment can be done using any popular social media which provides private messaging. Once the shared key established among different parties involved in the sharing, data is transferred using a different channel encrypted using the above mentioned key. Data channel can be any cloud storage provider or providers. The key idea behind FaceBox is, by reducing the visibility of a given provider towards the shared data the data owner can leave minimum trust on each provider and can still use to share data securely.

Main contributions of the FaceBox are, a) to provide secure file sharing among trusted parties over minimally trusted cloud serve providers and b) to securely consolidate available data space over different cloud service providers.

The rest of the document is organized in the following way. Section two will provide a background on related work being done upto now which is followed by the design concepts of FaceDrop. Implementation details are provided in section 4 while section 5 gives an alternative usage as a secure backup. Section 6 will provide a thorough security analysis. Future work is presented in section 7 and finally concluding remarks are given in section 8.

2. RELATED WORK

There are many threats to the security in cloud computing. Kaliski and Pauley [?] argue that features that define cloud computing such as automated transactions and resource pooling make traditional security assessments difficult. In some cases these features even cause security breaches. Majority cloud storage providers measure activity within its infrastructure. Kaliski claims that collection of this information is a security threat. Since FaceDrop leaves minimum trust on cloud providers, this issue is largely solved.

In their survey, Bhadauria et all [?] classifies threats into 3 groups: basic security threats (SQL injection and Cross Site Scripting Attacks), network security attacks(DNS and Sniffer attacks,using reused IP addresses, BGP prefix hijacking) and application level security threats(DoS attacks, cookie poisoning, concerns with hypervisor, hidden field manipulations). Similar work was done by Khalil et all [?] who identified 28 cloud security issues and classified these issues into five security categories (security standards, network, access, cloud infrastructure, and data). Bamiah and Brohi [?] highlighted seven seven deadly threats and vulnerabilities encountered in the clouds. They concluded that there is need to develop and design in-depth security techniques specifically for the online storage systems.

According to Cachin and Schunter [?], 175 data breaches have been registered in 2011 in the United States alone, that involved more than 13 million records. Amazon Cloud Service crashed during upgrade, hackers penetrated into Sony Playstation network and accessed users private data, and Dropbox temporarily allowed visitors to log in to any accounts using any password. These breaches further strenthen our claim that, cloud service providers can not be fully trusted with the user data and has to find a way to reduce the amount of trust we can place.

Kamara and Lauter [?] suggested to build a cryptographic cloud storage on top of a public cloud infrastructure where provider is not trusted by the user. In their approach users encrypt the data before the upload and then can create tokens to control who has the access to which data. They also offer "proof of storage" - protocol by which the server can prove the integrity of user's data. While this gives a good secured cloud storage for personal backups, for file sharing between two parties and reducing the visibility its not upto the mark.

Zhao et all [?] used a progressive elliptic curve encryption for creating secure cloud storage. User encrypt the data first before uploading it to the cloud. When he shares the data, data is re-encrypted without being decrypted. This re-encrypted data will then be cryptographically accessible to the authorized users only. They make a very strong assumption that the private key of the cloud provider is known to the data owner which is a bit of an unpractical assumption to be made in the current setup.

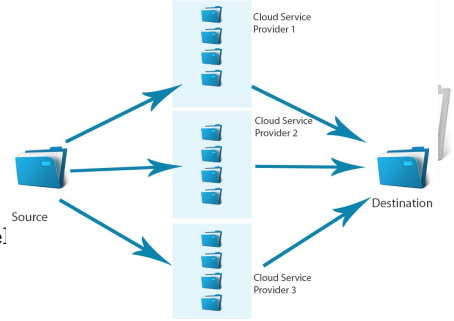

Figure 1: Face

Figure 2: File Sharing and Merging

Mulazzani et all [?] described several ways in which deduplication could be used to access files uploaded to Dropbox. If user A share hash value of his uploaded files with user B or B steals the hash value from A, B could download the files that belongs to A without any permissions from A. Moreover, when it happens, user A does not know it. This vulnerability again shows the poorly structured security mechanisms in placed cloud storage and the need for having a one extra layer of security only known to the end user.

In our project we look at the process of the data sharing from the user's perspective. The goal of this project to give a user more control over his data and laving minimum trust.

3. DESIGN

In the FaceDrop, the main user scenario considered is when an data owner wants to share data between trusted parties over the untrusted cloud storage providers. As it is mentioned above in the introduction the main goal in the design is to separate the authentication channel and data transfer channel so that for a given the service provider, the visibility towards the communication is minimum. This reduced visibility will further enhance the chances of protection in the event of a system wide compromise in the cloud provider.

3.1 Authentication

The Figure 1 shows the high level architecture diagram for the FaceDrop with the separated channels. Authentication channel in the FaceDrop could be anything with a pre-authentication i.e. a channel that requires the user to log in before initiating the authentication. To fulfill this requirement the solution is to go with a popular social media which allows the messaging between two (or group of) people.

FaceDrop uses the Diffie-Hellman [?] key exchange protocol to agree on a symmetrical key that is going to be used in the communication channel. It has been shown that Diffie-Hellman is prone to man in the middle attacks and the authenticated Diffie-Hellman [?] would solve it to the greater extent. For such authentication public keys infrastructure can be used. An user can bind a public key with the social media account being used so that parties involved in the key establishment can easily acquire the public key. Given the authenticated Diffie-Hellman is used, the array of different social media that can be used is significantly high. However in choosing the authentication channel one has to be little bit cautious as not to leave a public trace of communication that might have negative implications.

3.2 Data Channel

Once the key agreement is done data is shared using the data channel which can be a single data provider or span

across multiple data providers to which the data owner has access to. The parties receiving the data do not have to have signed up account in any of these cloud services to receive the data. Symmetrical key agreed upon using the authentication channel is used to encrypt the data. Thus only the parties involved in the authentication channel have the key for decryption.

To further enhance the security or to reduce the visibility, FaceDrop shards the data and distributes them between available could providers. The distribution can be random or round robin which ever agreed upon before initiating the channel. With sharding a given cloud service provider will only have a portion of data and this will make it further hard if the provider gets compromised or get evil to do any harm to the user data. This will also give the data owner consolidate available space for sharing data over cloud. Sharding will only happen if the data owner has access to multiple cloud service providers.

The data sharing also gives another added feature in FaceDrop. The default mechanism is to run the authentication protocol once and use the symmetrical key for the rest of the communication of that session. However design does not prevent an overly paranoid user to have different set of symmetrical keys for each chunk of data being passed using different cloud service providers. This will guarantee that even if the key being used in one provider is revealed, it will not affect the chunks pass on using other providers. This requires the parties to execute the key exchange protocol per provider involved in the sharing. However this will have a significant performance overhead.

3.3 Data Windowing

Data windowing gives the data owner the ability to transfer unlimited data to other parties. Data windowing in FaceDrop functions exactly same as it happens in TCP [**?**]. Data owner can specify the space FaceDrop can utilize in each provider which correspondents to window size in TCP. If multiple data providers are used, multiple window sizes can be defined. Window size can also be used to deal with parties having slow Internet connection. One caveat in this approach is everyone involved in the sharing has to be online at the same time to be able to transfer data. Though staged transferring can be applied where owner can transfer segments as and when other parties come online.

3.4 Client

In the FaceDrop architecture, each party involved in the data transferring has to run the client application locally. Client application will be responsible for data encryption and sharding at the data owners side and merging and decryption at the other side of the data channel. Client application will also carry out the Diffie-Hellman authentication for the users involved in the key exchange.

3.5 Data Initiation

Since key exchange protocol executes in the separate channel, there has to be a way to let other parties know about the starting point of the data channel. The same question remains same for sharing the links to download files (or encrypted chunks) with other parties. Different options available for initiating data transferring varies from sending the starting link through an email to using the same key exchange channel to send the link.

Figure 3: Diffie-Hellman over FB privat messeaging

To share the links of different data chunks, available options are embedding the link to the next chunk in the current data chunk it self or use the key exchange channel. Either way, data windowing is supported and all the links should be encrypted using the symmetrical session key.

4. IMPLEMENTATION

To this end, FaceDrop is implemented using Facebook [**?**], DropBox [**?**] and Google Drive [**?**]. Facebook acts as the main authentication (or key exchange channel) and DropBox and Google Drive function as cloud storage providers.

Authentication Protocol

Extensible Messaging and Presence Protocol [**?**] interface is used to implement the key exchange protocol using the Facebook Chat. To initiate the protocol, data owner will have to select the parties involved in the key exchange (parties upto 3 is currently supported). All the parties has to be online to complete the Diffie-Hellman protocol. Protocol executes as private chat messages among the selected parties. All the messages are BASE64 encoded [**?**]. Picture 3 shows how the messages are looked like in the FB chat interface. Once the key is generated from Diffie-Hellman it is hashed using SHA1 to get the final symmetrical key. AES-256 is used as the encryption algorithm in passing data over data channel.

Data Channel

Both DropBox and Google Drive provides a simple yet sufficient enough APIs [**?**, **?**] for third party applications. Client application of FaceDrop acts as a trusted third party to these providers and uses their Java based API for the interactions. When using the DropBox api, it only allows the FaceDrop application to access certain parts of the space restricting it access from other personal information.

Data Initiation Protocol

As it is mentioned above in the Design section, FaceDrop can support multiple data init protocols. Once the key exchange is done, data owner sends the link to the first chunk of encrypted data using the agreed session key over facebook to all the parties. Few of the other possible data initiation methods are mentioned in the future work section of the paper.

Once the initiation is done, next step is to send the links to remaining data chunks. Users can pick two options, either keep on using Facebook messaging to send the encrypted links or to have the link embedded in the data chunks itself. In the later case once a file is downloaded from the link sent by the owner, the final bytes (predefined) actually represents the encrypted link to the next data chunk. Thus data chunks will form a chain of links. In the later case, Facebook is only used to send the very first link to the data chunk. Owner will notify whether the chunks are randomly distributed among different providers or not. This will only occur if the owner has multiple cloud storage account to work with. At the client side this randomization has no impact towards functionality but this will affect the security as

randomization reduce the chance of a given provider keeping a continuous chunks of data.

In the current system every party involved in the data sharing has to be online during the authentication phase. However if the data size fits into the available cloud storage, owner can push the data to the cloud and recipients can download the links at a later time. If the available space is not enough data windowing needs to be activated which again requires all parties to be online.

Data Windowing

The user can specify the window size by specifying the number of bytes that can be used from each cloud provider and specifying the combination of the maximum size of a given data chunk and number of chunks to be included in a window. Data owner will specify the maximum data space for each used cloud provider while the other parties should specify the values for maximum data chunks and number of files permitted in the window.

Once the initial values are passed on, all the control messages are sent using the Facebook (or authentication channel). The control messages are the messages sent by the receiving end towards the data owner signalling the downloaded files and the messages sent by the owner to the other parities with the links to the new data chunks. At the current system, if the data windowing is enabled embedding links in the data chunks it self is not implemented. It just requires engineering work in implementing this feature.

5. SECURE BACKUP

Another feature that FaceDrop can support in this case is to function as a secure backup storage to the data owner. Although there are quite number of services giving encrypted storage in the cloud, the main differences in FaceDrop is the available storage consolidation among different cloud service providers. As it is mentioned in the above this consolidation has other security advantages as in a single provider will not have the visibility towards the backed up data.

In the event of sharing already backed up data in the FaceDrop, user still has to run authentication protocol with the other parties to get the session key. Once the session key is generated, it will be used to send the initial link to the receiving parties and to share the key used in the encrypting the stored data. This is required because key used in encrypting actual data chunks was not something agreed upon by all parties before the encryption and was generated solely by the data owner. And in such a situation, data windowing is not necessary since all the data is already in the cloud. The client application which runs at the data owners machine has the mappings to all the uploaded chunks are will gradually share the each chunks as the recipient progresses and once the download is completed shared links are removed.

6. SECURITY ANALYSIS

This section provides an analysis on the threat model of the FaceDrop highlighting key aspects of vulnerabilities and strengths.

6.1 Authentication

FaceDrop uses the authentication channel to both generate a shared key and to authenticate other parities involved in the sharing. Just as authenticated Diffie-Hellman uses public key infrastructure to tackle with man in the middle and at the same time tackle impersonations, FaceDrop uses the logged in Facebook account for the same reason. FaceDrop assumes that it is communicating with the legitimate parties, since logging into the Facebook account require credential that are supposed to be known to the assumed party. If the Facebook account is hacked by means of a virus or more sophisticated means, then FaceDrop could be communicating with the wrong person pretending to be someone else. Thus DropFace does not have the ability to deal with impersonation. Even with the use of PKIs it is known to be vulnerable to impersonations with forged public certificates.

The basic Diffie-Hellman is known to be vulnerable to man in the middle attacks as it is mentioned above. One advantage of using Facebook over conventional public networks is, Diffie-Hellman in FaceDrop runs on a one added security layer of Facebook communications. Thus if an adversary wants to sniff the Diffie-Hellman, Facebook secure communication has to be breached as well. This does not by any means guarantee that its impossible. It just makes the breaking in and sniffing on the Diffie-Hellman lot more harder. Thus reducing above mentioned threat to a greater extent.

While there are many products that gives encrypted storage in the cloud space, FaceDrop's authentication channel makes sure that only the parties involved in the key exchange have the key and the data owner does not have to share a symmetrical key by manually i.e. via a voice calls, emails, skype, etc as in with many commercial products.

6.2 Trusting the Untrusted

The most important question of most of the secure cloud storage and sharing solutions is to on whom should we trust and to which extent ? FaceDrop also requires data owners to trust certain elements in the cloud. And few elements to a greater extent than the rest of the elements.

Authentication channel carries a significant portion of data in terms of sensitivity of data. In FaceDrop, Facebook can see all the Diffie-Hellman protocol execution. Thus all the vulnerabilities found against basic Diffie-Hellman can now be exploited by Facebook (or by any other authentication channel provider). Although once the key is generated rest of the communication over Facebook private messaging is encrypted using the session key, the confidentiality of the Diffie-Hellman protocol relies on the amount of trust we place on Facebook. This can be further reduce by using Authenticated Diffie-Hellman by using a PKI. Even in the case of Facebook going evil, it has little knowledge about the actual data communication so then again the harm or the threat posed by Facebook is minimal.

The inability of trusting the cloud service providers is what motivated the FaceDrop. Cloud service providers usually encrypted data before storing in their storage in order to protect the data from outside intruders. However this does not guarantee any security against insider threats or even in the event of an outsider intrusion, where the adversary could get hold of keys used in encryption thus removing that security assurance as well. To this end, by using FaceDrop cloud service providers are completely blind on the keys being used to encryption. While this is being already done to some extent by others, in such services, in the event of a key exposure, all the data will be revealed since everything

is stored in one place. To contrary FaceDrop introduces further security by slicing the data into chunks and distributing them across different cloud service providers. Thus if key is exposed to a provider by any means, it will only do very little harm since only a random set of noncontinuous chunks are available to the provider.

The file slicing part would also help a different cause i.e. protecting against unlawful spying and data collection. With data slicing a given provider does not have full data at hand. Thus the cloud provider can give only so much of information which actually does not pose any threats to the data owner.

6.3 Protection Domain

FaceDrop is as secure as the local hosts are secured at the sender and recipient machines. If the data owners machine is compromised in a way that it can sneak into client applications memory then it can extract keys being used, mappings between file chunks, etc. Thus it can basically access anything of it's interest. Although the client application is password protected, there are many ways to break in. Thus the protection domain of FaceDrop is from data owners client application to the recipient's client application and it can not safe guard against any compromises in local machines.

7. FUTURE WORK

There are many enhancements can be done to FaceDrop to make it more secure and give extra features to the end user. Firs and fore most would be to support authenticated Diffie-Hellman using a Public Key Infrastructure. This would given extra bit of confidence in the authentication as it is mentioned in the previous section. Elleptic-Curve based Diffie-Hellman is known to be more secure compared to the other approaches by providing high entropy and such an addition would be very useful as well. Another addition would be to support more efficient group authentication to be able to share files between a larger group of people.

A service such as FaceDrop can actually be extended to provide a service of an authenticated URL shortener. While URL shorteners such as Google service and TinyBit URL provides shortened links to files, they do not provide authentication. FaceDrop could use features of cloud service providers to make sure that data does not leave the service and the authentication channel can be used to send unique links to a verified people. This will make sure only intended party views the data.

Securing data over the untrusted providers proved to be very hard and project like SPORC [**?**] have moved into different directions by detecting any wrong doings to the data being shared. Out of band channels such as the authentication channel can be used to verify the data being downloaded to make sure that data which are being downloaded are legitimate. Such an approach will increase confidence on the integrity of the data being shared.

Another interesting approach to initiate the data channel would be to use hidden channel of communication to send the initial data link. steganography can be used effectively to hide the link in a picture. One can imagine using a Facebook image to hide the data. The question to overcome here is to how to avoid the Facebook image compression or to have an approach that is compression resistant. Such an approach will further leave cloud providers in dark about the line of communication between the involved parties.

Another line of future work of this project includes the thorough performance and usability analysis. Such an investigation should look into the performance of the proposed approach for large files and multiple users. Different encryption algorithms and key establishment protocols can be evaluated against different user scenarios to come up with the most effective set of combinations.

8. CONCLUDING REMARKS

The original objective of the paper was to implement a secure file sharing platform on top of a minimally trusted cloud service provider. With that goal in mind, paper present FaceDrop which have separated channels to carry out the authentication / key exchange and actual data transferring. This separation as shown above let the user to place minimal trust in cloud storage providers effectively. FaceDrop is implemented using all publicly and freely available cloud services.

9. ACKNOWLEDGEMENT

Authors would like to extend their gratitude to Bill Aiello for introducing whole new set of interesting concepts in Security and for the continuous guidance throughout the class and during the project.