Shatter Secrets: Using Secret Sharing to Cross Borders with Encrypted Devices

Erinn Atwater and Ian Goldberg

University of Waterloo, Ontario, Canada
{erinn.atwater,iang}@uwaterloo.ca

Abstract. Modern consumer electronic devices such as smartphones and laptops are laden with intimate personal data such as past conversations, photos and videos, medical information, and passwords for services that contain information on our entire lives. This makes the devices of particular interest to law enforcement officials during even routine searches. A particular threat to users is when crossing international borders, as we have repeatedly seen reports that the data on these devices is subject to search and seizure without warrants or even suspicion of wrongdoing. In some cases, travellers have even been compelled to provide PINs, passwords, encryption keys, and fingerprints to unlock their devices.

In this position paper, we argue for the use of threshold cryptography to distribute encryption keys into shares, which are then securely transmitted to friends residing at the traveller's destination. When a traveller is subjected to scrutiny at the border, they are technically unable to comply with requests to decrypt their devices. Assuming the traveller is permitted to complete their journey, they must then physically interact with some (user-configurable) threshold number of their friends on that side of the border to recover their encryption keys. In our proposal, attackers must compromise both the traveller *and* a threshold number of the traveller's friends in order to learn anything about the secret key; the friends are unable to collude without the traveller present.

We also implement Shatter Secrets, an open-source prototype Android app aimed at realizing this goal.

1 Introduction

Crossing international borders in recent times has become fraught with uncertainty over the privacy and security of our electronic devices. Rather than merely the clothing and toiletries in our bags, our smartphones and laptops contain huge troves of intimate information, including photographs, financial and medical information, and correspondence, that often go back many years. In 2017, the United States Customs and Border Protection agency searched approximately 30,000 consumer electronics devices of travellers—more than triple the number of searches performed in 2015—and generated 250 complaints about warrantless searches [4, 11]. Even with the capability to use PINs, passwords, and disk encryption on these devices, travellers have reported being compelled to provide passwords, being asked to use their fingerprint to unlock smartphones, and having their electronics detained for extended periods of time while law enforcement agencies deploy forensic techniques against the devices [7, 11]. While some passengers refuse to comply with these requests (resulting in detainment of their device, their person, or being refused entry to the country [6]), certain international borders have even begun requiring large electronic devices be checked into the hold of the plane, removing the opportunity for the owner to refuse imaging and allowing for surreptitious inspection of the device and its contents [3]. Even the more restrictive guidelines on searches provided by the U.S. Department of Homeland Security in January 2018 allow access without a warrant to any data on the device that does not require a network connection [9].

In this position paper, we argue for using threshold cryptography to make it technically impossible to comply with such attempts to compel a traveller to surrender their passwords or encryption keys (the $5 wrench attack[1]). In our proposed system, the traveller does not *know* their encryption keys at the time of crossing the border and being subjected to security scrutiny, and so cannot be compelled to provide it even under threat of detainment or deportation. It is of course important that this fact be made very clear by the software itself, or possibly being well known via the popular media, so that it is incontroversial that the traveller is unable to decrypt the device, and no amount of threatening or arrest will change this fact.

By using strong device encryption in combination with our method of distributing the decryption key, attackers (including border agents and law enforcement) are unable to access the contents of the traveller's device even with coerced cooperation. This defence anticipates the event of being compelled to provide a password, and fails safe by protecting against data disclosure even when the defender's mind has been compromised.

2 Secret Sharing

Cryptographic secret sharing schemes [2, 12] take some arbitrary secret data D and divide it into $n \geq 2$ shares, with the intention of those shares then being distributed to n distinct parties. During the sharing process, a threshold t with $1 \leq t < n$ is chosen such that any subset of $t+1$ shares can be used to recompute the secret, but no subset of t shares reveals any information about the secret whatsoever.

Several others have proposed using threshold cryptography schemes for protecting data on users' personal electronic devices [1, 10, 13]. Our position builds on this work by proposing using these systems for the specific use case of crossing international borders and placing shares in the hands of the user's friends, instead of (just) their other personal devices. In the next section, we describe some of the modifications we make to account for the unique circumstances of the border-crossing scenario.

[1] `https://xkcd.com/538/`

3 Position

We propose using the following system to conceal encryption keys when attempting to cross an international border (or any other situation where the user anticipates being subjected to compulsion of their passwords):

0. Begin with a secret S, which could be an encryption key for a primary device or a password to a cloud service. In the latter case, it is up to the user to ensure they cannot be compelled to reset the password (e.g., via email).
1. Generate a symmetric encryption key K.
2. Choose a set of friends of size $n \geq 2$, and a threshold number of those friends t such that $2 \leq t + 1 \leq n$.
3. On a secondary device, use (t, n)-Secret Sharing to split S into n shares, and encrypt each share using K.
4. Send an encrypted share to each of your n friends (using a secure channel such as Signal or TLS); friends should import the share into an app that only allows exporting via NFC.
5. Erase S and all of its shares from memory on both devices; retain K.
6. Travel across the border (or other security checkpoint) with both the primary and secondary devices.
7. Upon safe arrival at the destination, visit $t + 1$ friends and tap their phones with the secondary device to retrieve their encrypted shares via NFC.
8. Decrypt each share using K, and use them to recover the secret S.
9. Decrypt the primary device or log in to the cloud service using S.

We employ the use of a secondary device as a convenience mechanism for implementation. Performing both encryption and recovery on the same device would require performing step 9 in a “bootstrap” area of the operating system prior to the device’s disk being decrypted, which, for example, would require rooting an Android phone to permit such a modification; however, in the event that the standard Android lock screen incorporates our required functionality, the secondary device would be obviated. We specify NFC as the transfer mechanism for secrets because it makes remote communication of the encrypted shares cumbersome; we do not want security agents impersonating the traveller and requesting their friends read out secrets over the phone, and we absolutely do not want them to be able to simply request the secrets be delivered over the network (even with a confirmation popup on the friends’ devices, many people are subject to security warning fatigue and will simply agree to such dialogs without authentication).

One concern is that security agents will image the encrypted contents of the primary device (possibly in secret) and the share decryption key K before allowing the traveller on their way. If the traveller then communicates shares over an insecure channel, they will be subject to interception and subsequent decryption of the primary device. By using NFC, we encourage the user to choose friends that are physically located at the travel destination (instead of, for example, choosing friends in their home country and attempting to communicate shares over the phone later). Another concern is that the initial transmission of encrypted shares might be recorded in global passive data collection if a secure channel is not used, which would permit security agents to retroactively recover the shares when an encrypted device is discovered. Transmitting the shares initially over a secure channel with perfect forward secrecy, and requiring physical interaction to recover the shares, mitigates these concerns. To compromise the entire system, such an adversary would have to compromise some subset $t + 1$ of the traveller's friends' devices to recover their encrypted shares in addition to the traveller's devices themselves.

Encrypting individual shares using K prevents $t{+}1$ friends from collaborating *without* the traveller to recover their secret (which could allow remote access to a cloud service).

Alternative approaches to solving this problem frequently include the traveller mailing the password to themselves, or downloading their data from a website (which possibly only comes online after a certain amount of time, or after friends have confirmed the traveller's arrival). We note that all of these approaches rely on lying to border agents (which we deliberately do *not* advocate for as part of this position paper), or on actions that can be easily impersonated (such as texting a friend), or on actions that the traveler can be compeled to perform (such as video-calling a friend). We note that all of these approaches rely on lying to border agents (which we deliberately do *not* advocate as part of this position paper), or on actions that can be easily impersonated (such as texting a friend), or on actions that the traveller can be compelled to perform (such as video-calling a friend). Another similar project to ours is Sunder,[2] which aims to allow people to use Shamir secret sharing in a usable manner. It does not, however, focus on the border-crossing scenario as our project does.

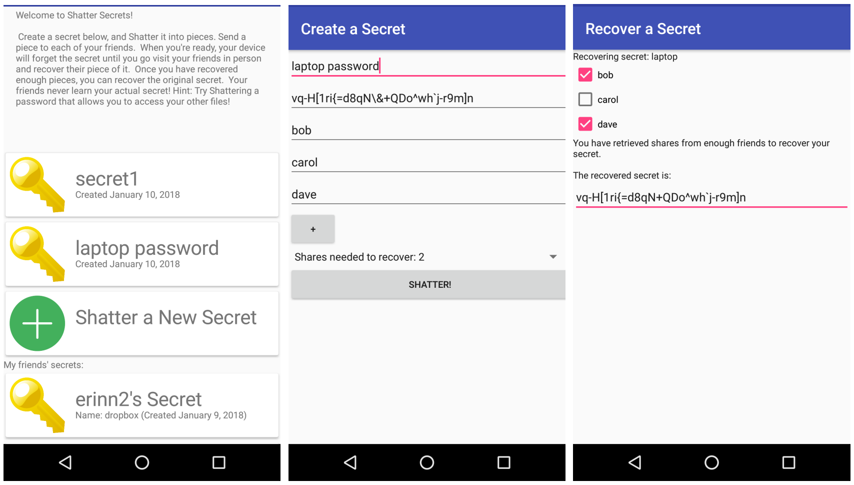
4 Implementation

We implemented a prototype of our proposal as an Android app called *Shatter Secrets*, shown in Figure 1. It is free and open source.[3] Users are asked to make an account on our server, which effectively acts only as a relay server for transmitting encrypted secrets over TLS. At registration time, the app generates a public-key encryption keypair and transmits the public key to the server, to be used for end-to-end encryption of encrypted shares being relayed to each designated friend. (Another option is to use Signal[4] disappearing messages to transmit encrypted shares; there is some precedent in Canada [5] and the United States [8] that text messages are considered private even when sitting on the recipient's device.) The user can enter arbitrary secrets, and the app will carry out the process described in Section 3. In our suggested configuration, the user installs Shatter Secrets on a secondary device, and uses an encryption key for

[2] `https://freedom.press/news/meet-sunder-new-way-share-secrets/`

[3] `https://crysp.uwaterloo.ca/software/shattersecrets`

[4] `https://signal.org/`



Welcome to Shatter Secrets!

Create a secret below, and Shatter it into pieces. Send a piece to each of your friends. When you're ready, your device will forget the secret until you go visit your friends in person and recover their piece of it. Once you have recovered enough pieces, you can recover the original secret. Your friends never learn your actual secret! Hint: Try Shattering a password that allows you to access your other files!

secret1
Created January 10, 2018

laptop password
Created January 10, 2018

Shatter a New Secret

My friends' secrets:

erinn2's Secret
Name: dropbox (Created January 9, 2018)

Create a Secret

laptop password

vq-H[1ri{=d8qN\&+QDo^wh`j-r9m]n

bob

carol

dave

Shares needed to recover: 2

Recover a Secret

Recovering secret: laptop

☑ bob
☐ carol
☑ dave

You have retrieved shares from enough friends to recover your secret.

The recovered secret is:

vq-H[1ri{=d8qN+QDo^wh`j-r9m]n

Fig. 1. Shatter Secrets running on Android, showing the list of created secrets and shares received from friends, the configuration screen for sharing a new secret, and a secret being recovered after retrieving shares from two of the three friends.

their primary device as this secret. Friends are selected by entering the usernames they registered in their respective copies of the app. Once a threshold value is chosen by the user, the secret is shared using Shamir secret sharing [12] and encrypted shares are sent to the relay server, to be pushed to the selected friends' devices. Encrypted shares are deleted from the server once they have been retrieved, and the user is informed when all of their friends have retrieved their respective shares and it is "safe" to cross the border. After crossing the border, the user must visit $t+1$ friends in person; each friend confirms they have authenticated the user in person by picking their secret from a list (as shown in Figure 1b), which will cause the app to then broadcast the encrypted secret via NFC. The friend's device forgets the encrypted share once it has been successfully delivered. When this process has been performed $t+1$ times, the user's copy of the app decrypts the shares and recovers the plaintext secret for them. If the secret was used for encrypting a primary device, or was a password to a cloud service, the user can then manually type it in on a separate device or app.

5 Conclusion

We argue that international border security agents have no business rifling through the intimate data stored on our personal electronic devices without a warrant or consent. We proposed using threshold cryptography to make it impossible to comply with such attempts on the spot. By distributing encryption keys amongst trusted friends at the traveller's destination prior to travel, the traveller cannot be compelled to provide access to their devices immediately. Instead, some subset of the trusted friends must be approached individually and compelled to provide their share of the key—a process which would hopefully invoke their rights against search and seizure as citizens of the country in question.

6 Acknowledgements

This work was made possible with funding from the Natural Sciences and Engineering Research Council of Canada Discovery Grant RGPIN-03858.

References

1. Atwater, E., Hengartner, U.: Shatter: Using threshold cryptography to protect single users with multiple devices. In: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. WiSec '16, New York, NY, USA, ACM (2016) 91–102
2. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the National Computer Conference. Volume 48. (1979) 313–317
3. Calder, S.: Security experts astonished by electronics ban on Middle East airlines. The Independent (March 2017)
4. CBP Public Affairs: CBP Releases Statistics on Electronic Device Searches. U.S. Customs and Border Protection (April 2017)
5. Connolly, A.: Text messages can be private once received, Supreme Court rules. Global News (December 2017)
6. Cope, S., Kalia, A., Schoen, S., Schwartz, A.: Digital Privacy at the U.S. Border. The Electronic Frontier Foundation (March 2017)
7. Fox-Brewster, T.: Feds Walk Into A Building, Demand Everyone's Fingerprints To Open Phones. Forbes (October 2016)
8. Johnson, G.: Justices: People have right to privacy in text messages. Komo News (February 2014)
9. Kopan, T.: DHS issues new rules for searching electronic devices at the border. CNN (January 2018)
10. Peeters, R.: Security Architecture for Things That Think. PhD thesis, KU Leuven (2012)
11. Savage, C., Nixon, R.: Privacy Complaints Mount Over Phone Searches at U.S. Border Since 2011. The New York Times (December 2017)
12. Shamir, A.: How to share a secret. Commun. ACM **22**(11) (November 1979) 612–613
13. Stajano, F.: Pico: No more passwords! In: Security Protocols Workshop. Volume 7114., Springer (2011) 49–81