Dark Crystal - Briar Project Case Report

Reflections on Designing, Implementing & Testing Two Features in the Briar App using Dark Crystal

Magma Collective

Contents

1 Introduction

As an initial case project for the Dark Crystal protocol and developer toolkit, two features were implemented for the security-oriented messaging app 'Briar', 'Social Backup' and 'Remote Wipe'.

'Social backup' allows users to backup and restore their Briar account and contact list using a small group of trusted contacts who collectively store the backup data. This backup data is distributed in such a way that a critically large subset of the trusted contacts are needed in order to restore the backup, meaning there is a tolerance to some group members being unavailable.

'Remote Wipe' allows users to assign a small group of trusted contacts the ability to remotely trigger the deletion of their Briar account. It is designed for situations where the user's device is assumed to have been physically compromised, for example because the user has been arrested or captured along with their device.

This report is divided into two sections - developer oriented and user oriented. The first section reflects on the utility of the developer toolkit in the context of this case project and proposes improvements to the documentation and Java libraries for the benefit of future projects using Dark Crystal.

The second section reports on the user-testing sessions which took place after the initial development iteration, discusses the utility of the features implemented, and proposes improvements to the features for a further development iteration.

2 Developer toolkit evaluation

The design phase began with a series of meetings with Briar's developer team where we looked at the protocol and sketched out the back and front end components of the Social Backup feature.

The design process is discussed in a series of blog posts.

2.1 Existing security features in Briar

Briar's existing security features meant many of the recommendations in the Dark Crystal protocol were already present, such as cryptographic signing and end-to-end encryption of the protocol messages. This changed considerably how we used the Dark Crystal libraries, and meant that only the low-level libraries were needed.

Perhaps this makes Briar a bad candidate to examine the use of the Dark Crystal developer toolkit, since we are able to leave a lot of the heavy-lifting to the underlying protocol. On the other hand, since Briar is designed specifically for high-risk users, we are required to be extra-vigilant when it comes to security.

2.2 Serialisation Protocol used

The Dark Crystal protocol recommends Google Protocol Buffers as a serialisation protocol for the messages, wheres Briar uses a custom made protocol 'Binary Data Format' (BDF).

To stay consistent with Briar's existing conventions and avoid introducing an extra dependency, BDF was used for the social backup and remote wipe features. BDF has two basic data structures, `BdfList` which extends `ArrayList` from the Java standard library, and `BdfDictionary` which extends `TreeMap` from the Java standard library, and takes strings as keys. Most of Java's primitive types can be encoded, and each type has an additional 'optional' form which allows null values. BDF has all the features needed and is very intuitive to use. But since it is not released as a standalone library, we will continue to recommend using Protocol Buffers to other projects.

- BDF Specification

2.3 Packaging of the native dependency

We hit a difficulty when integrating our secret-sharing library into Briar. Briar's back-end is designed to be platform independent, to make it possible to build clients for platforms other than Android. Our library contains bindings to C code, and the Android SDK toolchain is used for building the native part of the library specifically for Android. This means it is packaged as an Android Archive (AAR) rather than a standard Java Archive (JAR).

We found a work-around to fix this, but it made things more complicated as we need to have two versions of the library for different platforms.

2.4 Social backup

2.4.1 Separation of 'shard' messages and encrypted backup messages

The Dark Crystal protocol recommends bundling these messages together into a single message. But Briar's developers wanted to be able to dynamically update the contents of the backup without needing to re-issue shard messages. This turned out to be a great improvement and opened a lot of possibilities. We propose updating the protocol to make this standard.

2.4.2 Handling mismatched or invalid shards

Dark Crystal's proposed technique for validating the integrity of shards is using signatures. To validate a signature we need to know the public key of the author. So our system relies on the public key being

made public, or at least being somehow obtainable at the point of recovery. For example, GPG/PGP email encryption software often includes the option to publish a public key to a public key server, making it easy to retrieve the public key associated with a known email address.

With Briar, this is not the case, as no information is ever published anywhere publicly accessible. So it is difficult to retrieve your public key in order to validate a shard.

This problem is to some extent mitigated by enforcing the secret owner and custodian to meet in-person to return the shard. Assuming the custodians themselves are trustworthy, and since the custodian has already validated the signature of the shard message, there is little opportunity for a 'person in the middle' attack during an in-person exchange. However, in the future we would like to be able to offer the possibility to return shards remotely, and this issue would need re-considering for that to work.

Besides validating the integrity of shards, we can also check that all the returned shards belong to the same set. This is very easy to do, since the shards contain an identifier unique to the set. What is less clear, what is the desired behaviour when they do not all match.

Checking for mismatched shards was not implemented in the initial development round, and issues with this are discussed in the user-testing section.

2.4.3 Technique for returning shards

Returning shards to the secret owner following account loss is the most vulnerable part of the process from a security point of view, as it is difficult to authenticate the secret owner when they have a new cryptographic identity.

For this reason Briar's developers wanted to only allow shards to be returned by an in-person exchange initially. A remote shard exchange could be possible in the future, but we deemed it beyond the scope of this project. The security requirements for this, and a possible design, will be discussed in our final report.

Briar already has a protocol for exchanging data using a technique which guarantees a secure in-person transaction. It is called 'Bramble QR Code Protocol' (BQP) and is used for adding contacts nearby. Rather than re-invent the wheel, we planned to adapt this protocol to use it for shard return.

However, when attempting to do this, we encountered a problem. BQP is designed to be used when signed into Briar, and relies on Briar's transport plugins which in turn rely on the database being present in order to store and retrieve transport properties, as well as other services which are only initialised after signing into Briar. When returning shards, the custodian is signed in and can access their database, but the secret owner is not and will only initialise the database once the account is recovered.

This unfortunately meant we needed to write a similar protocol with from scratch, implementing our own networking code. This had the advantage that we could make the process simpler for the user. BQP is designed for a mutual key exchange and requires both parties to authenticate using a QR code. But when returning shards, data is only transmitted in one direction (from the custodian to the secret owner). So we only need one QR code to be scanned.

However, BQP can use both LAN over wifi as well as bluetooth. Because of time limitations, we only implemented shard return over LAN.

- BQP specification, version 4

Shard return is undoubtedly the most vulnerable part of the Dark Crystal protocol from a security perspective. Generally, we always recommend 'out of band' contact with the secret owner to confirm their new cryptographic identity. That is, the custodian and secret owner should communicate by some other means than via the application itself in order to confirm the identity of the secret owner.

In Briar's case, this involves meeting in-person, which makes it very hard for an imposter to impersonate the secret owner, but might be impractical, time-consuming, dangerous, and may well involve communicating by some other means in order to arrange to meet physically.

2.4.4 Backup contents

When we talk about backing up the 'Briar account', we mean backing up the keys which make the account 'yours', meaning your contacts will be able to continue to contact you.

But there are other kinds of data stored by the Briar application which peers might not want to loose, for example their contacts list, their messages, and their membership of particular groups, forums and blogs.

When designing the feature and deciding what to include in the backup, there is a trade-off of what information peers are most concerned about loosing, against what information they are most concerned about an attacker having access to should our backup become compromised. The process if deciding what to include in the backup for Briar, gave some insights useful to other projects which we propose to document in the developer toolkit.

Firstly, the content of forums or blogs themselves do not necessarily need to be backed up, since existing content can be retrieved from other peers when re-joining, and could contain sensitive information which would implicate others.

Our initial research showed high-risk users were most concerned about loosing their contacts list, as it can often be very impractical or dangerous to regain contact by other means.

Message content however, seems less appropriate to back up, as it is likely to contain sensitive details, and the other party of the conversation generally still has access to it. That is, re-gaining contact with

someone gives you indirect access to your previous conversation with them, and all you really gain by including the messages in the backup is the convenience of having them displayed on your device without needing to ask.

Besides the clear disadvantage with regards to security, backing up content also has the disadvantage of being arbitrarily big. Since we are asking others to hold the backup on their own devices, it becomes less practical appealing for them the bigger the backup gets.

So this feature provides a backup of the cryptographic identity securing the account, the contact list, and any backup shards held for other contacts - which will be discussed below.

2.4.5 Mutual inter-dependency by backing up shards held for others

As mentioned above, a major change to the Dark Crystal protocol we made with this feature was to separate 'shard' messages (the shares of the secret) from the encrypted secret itself. This allowed additional information to be dynamically added to the encrypted backup, for example when a new contact is added to the contacts list, the backup is updated to include them.

This made it possible to also add shards held for others to the backup. So when someone receives a shard, if they themselves have a distributed backup, they add that shard to it and send their updated backup out to their own custodians.

This means that when an account is recovered, so are the shards previously held by that peer. This makes the system very robust, as not only is there the threshold mechanism which tolerates some degree of loss of shards, but when shards are lost there is the possibility to recover them again.

2.4.6 Dangers of having the same identity on two devices simultaneously

One of the difficulties with distributed system architectures is resolving problems caused by two 'copies' of an identity. By allowing the backup and restoration of a Briar account, we make it possible that there are two copies of the same account on two different devices. Briar is not designed for use on multiple devices and this is likely to cause problems. The intended use-case for the recovery feature is for when access to an account is lost, either because the password is forgotten, or because the device is lost or damaged. But there is no way to prevent the account being recovered whilst it is still active on another account.

This problem is not unique to social backup, or the Dark Crystal protocol. Any way of backing up and restoring an account, for example making a local backup on an SD card, would introduce the danger of having both the original account and the restored version active at the same time.

In the initial development iteration this issue was not addressed. It is assumed that the feature is used as intended.

2.5 Remote Wipe

The idea for the Remote Wipe feature came from Briar's developers, and although it uses much the same principle as our protocol, of a critically-sized subset of trusted contacts having a particular ability, it does not require a secret sharing algorithm since it is possible to authenticate the trusted contacts by other means. So the technique has a lot of the same social implications as with social recovery, but technically it operates in a very different way, and is much simpler to implement.

Although it doesn't use the secret sharing library, the setup process for remote wipe is very similar to that of social backup. Both involve the user choosing a set of trusted contacts, sending messages out to those contacts, and the contacts being able to perform an action based on such a message.

When we were implementing the feature, it began to look so similar to social backup that we considered rolling the two features into one, with a single set of trusted contacts for both social backup and remote wipe, and with shard messages implicitly acting as remote wipe setup messages. However, we decided it was better not to do this without considering the implications, so we kept the two features separate, and used the user testing sessions to discuss the idea of having a single set of contacts for both.

2.5.1 Fixed threshold

Unlike social backup, remote wipe might need to be activated very quickly, in an emergency situation. Because of this, we decided on a fixed threshold of two, rather than allowing the user to choose a threshold themselves.

2.5.2 Revoking remote wipe status

Similar to with social backup, in the initial implementation phase we did not provide a way to revoke a remote wipe setup message. So once a contact is chosen to be a 'wiper', the will always be. However, unlike social backup, revoking remote wipe messages does not introduce complications with dealing with multiple sets of secret shares. So it should be very simple to implement later.

2.5.3 Time limit on wipe signals

An issue which needed to be addressed, is what to do in the case of a false alarm, or accidental sending of a wipe signal. We decided to add an expiry date to wipe messages, so that if a second signal is not received before the first signal expires, the first signal is deemed invalid. We initially set this expiry limit at 24 hours. This means if a user accidentally sends a remote wipe signal, they don't need to do anything to cancel it. As long as no-one else sends one that day, the signal will be ignored.

3 User testing report

3.1 Goals of the test sessions

- Asses whether the features give the behaviour intended.
- Asses the extent to which participants are able to use the features with only minimal guidance.
- Establish participants' comprehension and perceived utility of the features.

3.2 Format of the sessions

For each feature, two sessions were run, which we call session A and session B.

We had originally planned to hold the first round of sessions at an Internet Freedom community conference or event, such as RightsCon, the Internet Freedom Festival, or the Chaos Communications Congress. This would allow us to get feedback from security trainers, activists or other workers in the field who have an insight into the concerns of high risk users. Unfortunately due to the Coronavirus pandemic, all such events that we know of were being held remotely. We wanted to hold the sessions in-person, partly because the feature requires that the devices are physically nearby, and partly to get better qualitative feedback from participants.

So both sets of sessions took place in-person in small groups according to local restrictions. We used our social networks to find participants and when selecting potential participants we tried to get a diverse range of levels of technical experience. Some participants were selected because we considered them to have an understanding of the needs of high risk users, for example because they had a background in political activism.

3.3 Precautions taken

3.3.1 Not using own devices

Since these are unreleased features with potential security flaws, it is important that it is not used for 'real' messages. For this reason, devices were provided for the test sessions pre-installed with the feature, rather than participants installing the app on their personal device. Furthermore, the process of installing an unreleased package can be time-consuming, and time in these sessions is valuable.

In some cases, we wanted to test the app on particular hardware (for example, a tablet with bigger display dimensions). In this case we did allow participants to install the app on a personal device, but were sure to uninstall it after the session.

3.3.2 Isolated network

In all cases, a new account was created specifically for the test session, and contact was only made with other participants of the session, essentially created and isolated test network.

3.3.3 Confidentiality of logs and crash reports

In some cases, a bug was noted where it was useful to obtain program logs, and in one case there was a run-time crash which produced a crash report. Briar has a built-in feature to anonymously and securely submit these reports to their organisation. These were then forwarded to us by encrypted email.

3.3.4 Trigger warnings

When discussing potentially traumatic themes, for example being arrested or captured, we warned participants beforehand and got consensus as to whether to proceed with the discussion.

3.3.5 Language and terminology issues

All sessions were held in English, and all participants were proficient English speakers, although many were not native speakers. Although Briar is available in a range of languages, the interface components tested are currently only available in English. Because of this, extra time and care was taken to try to prevent language issues being the source of the problems uncovered.

In this report as well as our other documentation we use specialist terms which are not widely understood. Below we outline which terms we avoided using during the sessions.

3.4 Social Backup

Figure 1: Photo from session A

- Guided walk-through of the feature with installation instructions.
- Session A notes and photos
- Session B full internal report

3.5 Terminology

In this report we use the terms ‘trusted contact’ and ‘custodian’ interchangeably. Similarly the terms ‘backup piece’ and ‘shard’. During the session, the specialist terms ‘custodian’ and ‘shard’ were never used. Care was taken to use the same terminology used in the user interface of the feature.

3.6 Ice-breaker activity

Figure 2: Photo of ice-breaker activity, session A

To get to know each other and begin to think about trust, participants were asked to imagine they were going away for a week, and want to leave access for others to feed plants, pets, etc. They each described what their strategy would be with the keys to their home. They then were asked to discuss an equivalent process for digital files.

3.7 Test process

Participants ran through the backup and recovery process several times, taking different roles (either secret-owner or custodian). Initially, each participant played only one role. After all participants had played both roles in isolation, we had a round where all participants played both roles at once. That is, they all mutually backed up each other's accounts.

In session B, the feature was additionally tested using a tablet rather than a smartphone, which has a different display resolution and hardware features, to see how this effected the functionality and UI.

3.7.1 UI issues

All users were able to find the 'Social backup' settings option when asked to backup their account.

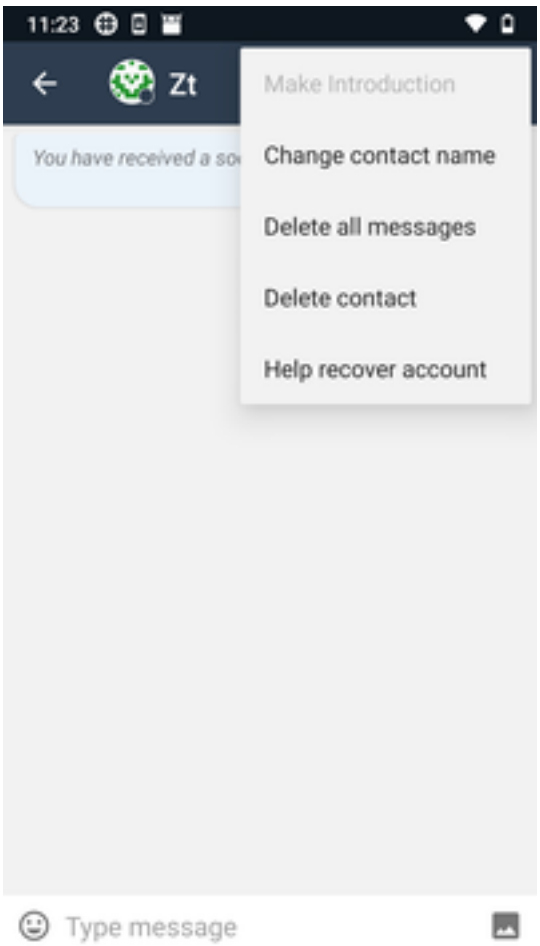

Figure 3: Screenshot of 'Help recover account' menu option

40% of users said the 'Help recover account' option was difficult to find (although all were able to find it).

Interestingly, in some cases the users with experience in technical jobs took the longest time to find both of these options.

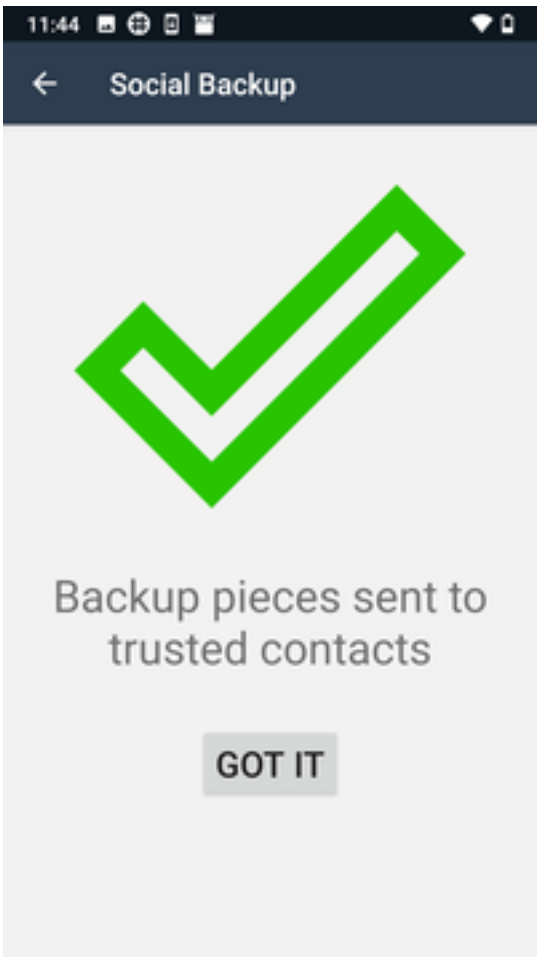

Figure 4: Screenshot of backup pieces sent notification

The ‘Got it’ button on the confirmation screen was confusing - some participants thought it was something to do with them confirming that that had received something themselves. It was suggested that it been changed to ‘Ok’ as this is more widely used and understood in this context.

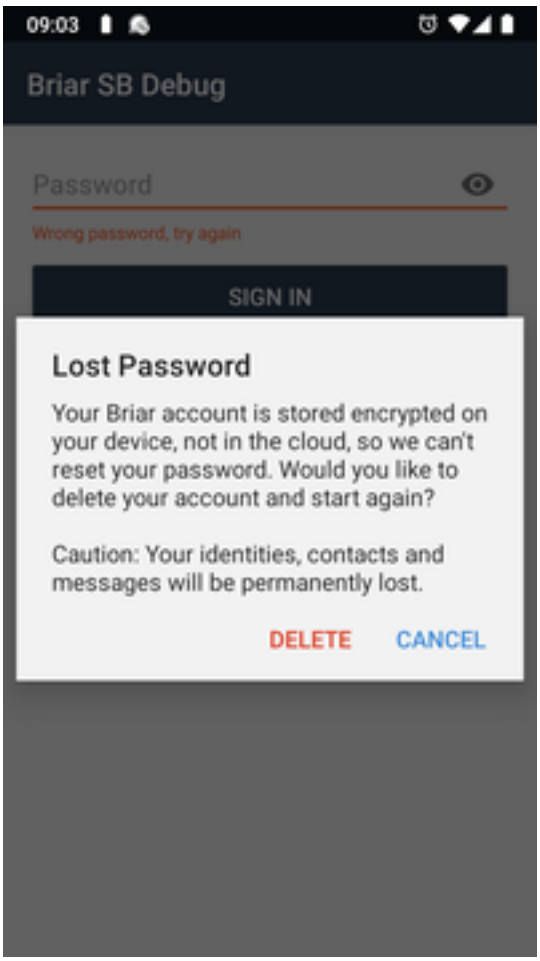

Figure 5: Screenshot of lost password explainer

It was noted that the 'Lost password' dialog (displayed when choosing 'I have forgotten my password') needs to be updated to explain about social backup recovery.

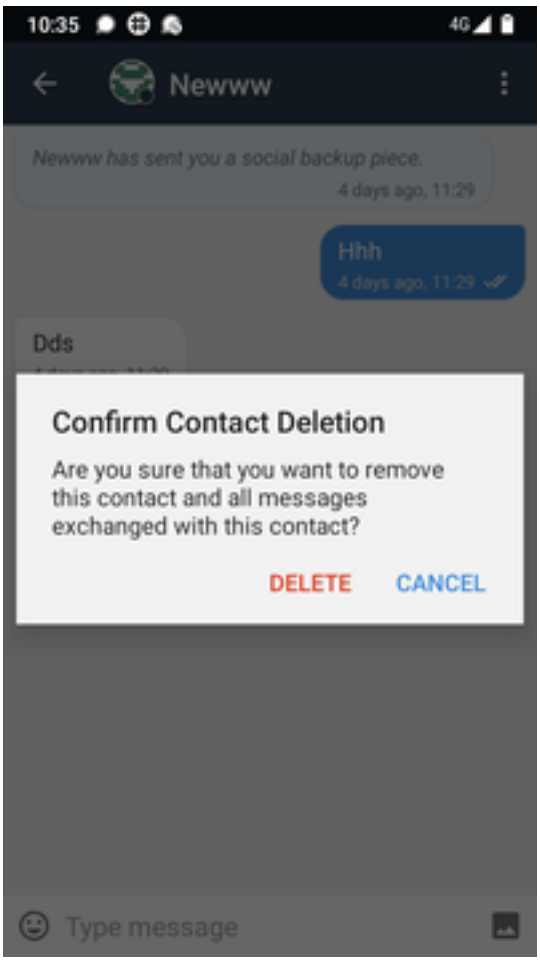

Figure 6: Screenshot of delete contact confirmation dialog

When deleting a contact who is a custodian, the ‘are you sure?’ dialog does not explicitly warn you that deleting this contact will mean that updates to your social backup will no longer be sent to this contact. This does not necessarily hinder recovery, since when recovering, the latest available version of the backup is always used. But it should anyway be made explicit a custodian is going to be removed.

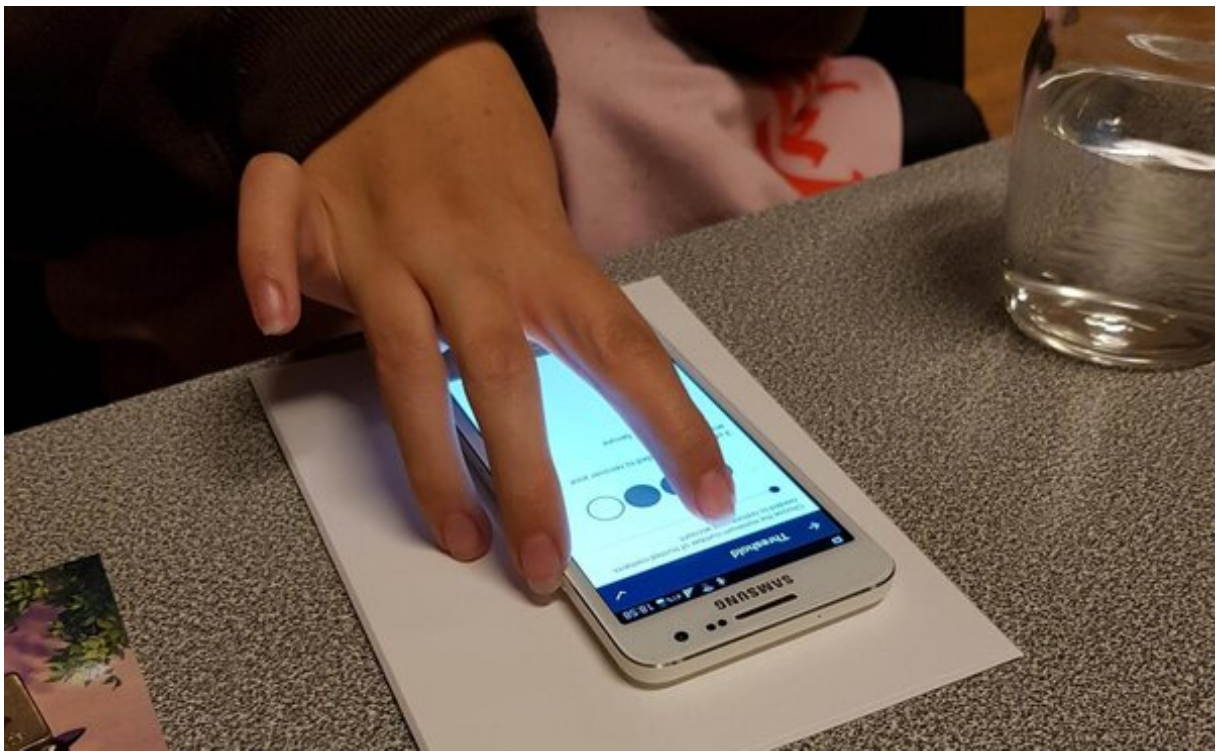

Figure 7: Photo of the threshold selector

The slider for selecting threshold was generally usable and understandable. One participant had difficult finding the button to confirm the selection (a tick icon in the top right corner of the screen).

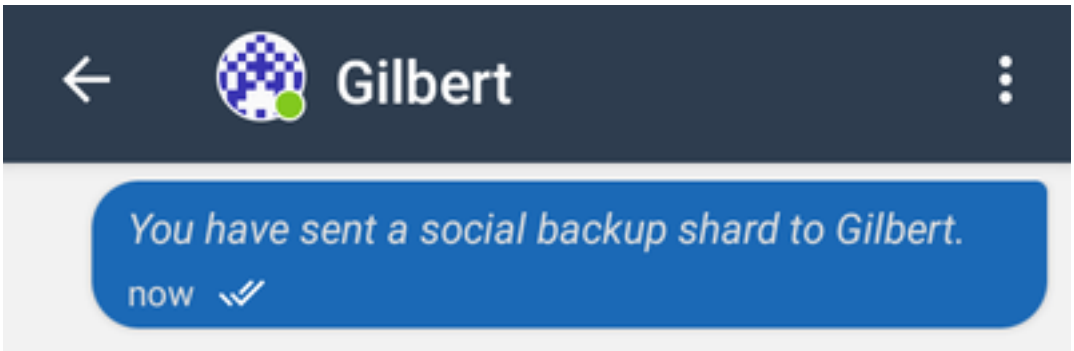

Figure 8: Screenshot showing confirmation tick icons

3.7.1.1 Confirmation of backup piece receipt Participants wanted to be sure their backup pieces had arrived with their custodians. 4 of 5 participants were able to do this by looking at the conversation with each custodian, and checking the icons next to the notification that a backup piece has been sent.

A clock icon means not yet sent, one tick icon means sent and two ticks mean confirmed as received by the contact. This is similar to the interface in popular mobile messaging apps. (add screenshot). The custodians appear at the top of the contact list screen, but it was noted that it would be more convenient to see an overview of confirming reception of backup pieces rather than having to check each custodian individually.

3.7.1.2 Not being able to change the group of trusted contacts 60% of participants said it would be desirable to be able to change which contacts were custodians, or create a new backup with a different set of custodians.

3.8 Bugs noted

3.8.1 Message delivery after recovery

Currently, after recovery, messages to existing contacts are never delivered. This issue was known before the test session and a fix is planned.

3.8.2 Contact deletion

When deleting contacts, the contact is not always removed from the contact list following a confirmation of deletion. In one case, Briar crashed when deleting a contact.

3.8.3 Mismatched shards

If shards from two different sets are given to the secret owner, it is impossible to recover the account. There is a danger that a custodian chooses the wrong contact before selecting 'help recover account', meaning shards from incompatible sets get muddled together. With the current implementation, once the secret owner has a single shard from the wrong set, they are unable to recover their account no matter how many shards from the correct set they have.

This became an issue during the testing session, after several rounds of trying the backup and recovery process,some participants had more than one contact for a single person in their contacts list. This is a familiar situation - when somebody gets a new phone number and you often end up storing two contacts in your phone for the same person, and cant remember which one is the current one when you want to call them.

As well as fixing this issue so that extra shards from mismatched sets are ignored, it was suggested that we could send the contact name together with the shard data, and display it to the secret owner,

making it clearer to them if the wrong contact had been chosen. This needs consideration as to whether it effects security.

3.9 Issues using a tablet rather than smartphone

The tablet used was a Huewei Mediapad AGS2-L09 running Android 8.0.0.

There were no additional issues regarding the user interface layout on a bigger display.

There was a difficulty when adding contacts with Briar's 'add contact nearby' feature, which involves a QR code scan and a handshake over Wifi or Bluetooth. However, we were unable to reproduce this problem when trying again with the same device after the test session.

3.9.1 Danger of using the same identity on two devices simultaneously

Unfortunately, due to the issues with handshaking with contacts following recovery, we were not able to use the sessions to investigate what problems occur when an account is recovered whist the original version of it is still active. This will have to wait until the second development iteration.

3.9.2 Proposed feature improvements

This is a collection of possible improvements based on feedback from both sessions. They are ranked by importance divided by estimated time needed and are implementing as many as we can in the timespan of our second development iteration.

3.9.2.1 Back end

1. Fix bug with deleting contacts
2. Transport key exchange after recovery
3. Failure to recover on mismatched shards
4. Persistent storage of returned shards
5. Fix setting default slider position on choosing threshold
6. Improve error handling during shard return
7. Improve handling of invalid/corrupt shard messages
8. Shard return over Bluetooth
9. Remote shard return - not possible in the next development round due to time constraints

3.9.2.2 UI

1. Change ‘Got it’ button used on confirmation screens
2. Change wording on shard sent confirmations (remove word shard)
3. Disable ‘help recover account’ menu option for contacts who have not sent you a shard
4. Improve ‘Lost password’ dialog.
5. When setting up a backup, if you have less than two contacts, do not display the threshold selector screen as it is impossible to choose enough contacts
6. Initial explainer screen on setup before choosing custodians
7. Improve screen showing existing social backup, and add delivery confirmation of backup pieces
8. Generally improve explanations on explainer screens
9. Notification/visualisation when recovering shards held for others
10. Add an extra warning when deleting a contact who is a custodian
11. Notification on backup updates
12. Improve visualisation of threshold settings
13. Make ‘help recover account’ an option in settings menu - decided against this

3.10 Mutual interdependence of social backups

An important change we made to the Dark Crystal protocol when integrating it to briar, was that we did not treat the shards and encrypted backups as a single message.

Briar developers wanted to be able to dynamically update the backup payload, because it contained the contact list, which changes over time as more contacts are added.

So we decided to have one-time shard messages, which were shards of the key to decrypt a ‘backup’ message, and many ‘backup’ messages which are incrementally updated versions of the encrypted backup itself. When a backup message is received, its version number is checked. If it is greater than the version number of the backup currently held, the old backup is discarded and replaced with the new one.

Putting this system in place gave us the opportunity to easily add other kinds of data to the backup at a later stage. Most significantly, we add the shards one receives **from other people** to our own distributed backup.

So when you get a shard from someones else’s social backup, if you have a social backup yourself you add the shard to that backup, and send out the new version to your custodians. When you recover your account, you are still holding the shard. This creates mutual interdependence and makes this backup system very robust.

3.11 Remote Wipe

Figure 9: Photo from session B

- Guided walk-through of the feature
- Session B full internal report

3.11.1 Terminology

The terms ‘wipee’ and ‘wiper’ are used in this report for brevity. As these terms are ambiguous and confusing, they were never used in the session, and the roles were more explicitly described as ‘the person having their account wiped’ and ‘one of the trusted contacts helping to activate the wipe’.

3.11.2 Ice-breaker activity

After a 'trigger warning', and consent to discussing a potentially traumatic theme, participants were asked to get into the role of a political activist organising and attending a demonstration. They described their possible concerns and fears and what measures they might take to stay safe.

3.11.3 Test process

During the remote wipe sessions, we tested the remote wipe feature in isolation, and then tested it together with the social backup feature, recovering an account which was deleted using remote wipe. The intention was to see to what extent the two features complement each other, and whether process of setting each of them up can be improved. We then opened a discussion relating to the social implications of the two features.

3.11.3.1 How social backup and remote wipe complement each other The two features we implemented in Briar have some big technical differences, but they both essentially use the same principle - of assigning a special ability to some critically large subset of a trusted support group. The two features complement each other because social backup can be used to restore the account following a remote wipe. This takes the 'dangerousness' out of remote wipe, making the 'wipers' more ready to use it. This might mean they use it even when there is only a suspicion that the wipee's device has been compromised, which makes the feature more powerful.

When we were implementing the user interface for remote wipe we realised how really similar the two features are from the perspective of users, and asked ourselves if it would actually make more sense to 'roll the two features into one' at the user interface level, making custodians for social backup and wipers be automatically the one and same thing. This would reduce the need to set up both features, saving time and cognitive load. It also addresses the problem that a user might not feel the need to setup a remote wipe until its too late - making it implicit in setting up a social backup means it is automatically there if needed. But the question remains - are there some situations where you would want particular contacts to be custodians of your social backup, but not able to activate a remote wipe, or vice-versa?

So we opened a discussion about this with participants after testing both features together, in order to address some of the security implications of the double-roles. These included:

- What happens if members of the support group have their devices compromised?
- Would security be significantly improved with two distinct support groups for each of the two features?
- Is more trust needed for one over the other?

- Are different thresholds appropriate?
- What are the implications of support group members knowing who each other are?

Generally a common theme in these discussions was what we will call ‘mobility’ vs ‘stability’. For social backup, people with stability would be preferable, such as non-activists or older family members, who might have a more stable home, move around less and so be less likely to loose their device with the backup shard. For remote wipe, people with mobility were preferable. More likely to be present in high-risk situations and have an understanding of the potential dangers, and potentially faster to respond.

One participant told a personal story which is relevant to the ways these features could be used. They are a prominent figure on social media, and were planning to travel over a border where they expected possible problems. In case anything should happen at the border, they wanted to be able to alert their contacts and make it public on social media. To do this they set up a ‘safety team’ who had access to their social media accounts and could inform their contacts and followers if they were to be abducted. The question of revocation came up, as they fell out with one member of their safety team and no longer wanted them to have access to their accounts.

3.11.4 General feedback

Generally, the feedback from the remote wipe sessions was very positive. Both the principle of it and the user experience were well received. Because of this, we propose recommending such a feature in the Dark Crystal documentation, and plan to explore other use-cases for it in our final report.

3.11.5 UI Issues

Figure 10: Remote wipe explainer screenshot

- Most participants did not understand that two remote wipe signals are required to activate the wipe. This was not made clear enough in the UI, both from the perspective of someone setting up the wipe and receiving the wipe.
- Regarding the 24 hour expiration time of the wipe signals, 4 participants thought this was appropriate, one thought it should be shorter, and suggested 3 hours.

3.11.5.1 Unable to revoke wiper status Two participants expressed desire to be able to change which contacts are wipers.

3.11.5.2 Needing to choose a set of trusted contacts for both social backup and remote wipe

Generally it was agreed that the social backup custodians and the wipers could be one and the same set of contacts. One participant described a situation where they might have a lot of trust in someone but not want them to be able to activate a remote wipe. They gave the example of an older relative with little technical experience, who might accidentally activate a remote wipe. They said this could be mitigated if other members of the support group were more technically competent, making it very unlikely that two wipers would activate a wipe accidentally.

3.11.6 Bugs

- Similar to with social backup - there were issues when deleting contacts who had been added as wipers. Although it was possible to delete these contacts completely using the 'delete contact' option, the 'delete all messages' option failed to delete the wipe setup notification messages.

3.11.7 Proposed feature improvements

3.11.7.1 Back end

- ☐ Make it possible to revoke wiper status - sending and processing 'revoke' messages
- ☐ Change wipe message expiration time

3.11.7.2 UI

- ☐ User interface for revoking wipes
- ☐ Change set of trusted contacts
- ☐ Add explainer screen during setup
- ☐ Improve explainer screen during wipe activation
- ☐ Combine with social backup, having one trusted support group for both features

4 Conclusion

Social backup is a very complex feature and introduces many possible difficulties, but when implemented and used correctly can be a very empowering mechanism, giving users a way of ensuring the integrity of their data whilst maintaining ultimate control over it. This makes it an appealing option for high-risk users who need to be able to operate independently of a service provider. There is however still work to be done to improve usability and address security issues.

Remote wipe is more simple and robust. It is much easier to get right, but it is only useful in quite specific circumstances. Its utility depends on the types of risks users might face, but since it is not so difficult to implement and use, it could be worth implementing even if it would be used very rarely.

The concepts involved in social backup and remote wipe were new to most participants of the testing sessions, but well received and understood.

5 Appendix - Materials used for user testing

5.1 General questions for participants

These questions were asked to participants to get an idea of their background and experience with security tools:

- Are you familiar with using Android?
- Do you own an Android device? If so, since how long?
- Have you worked in a job you would describe as 'technical' in the last five years?
- Do you use email encryption? Have you heard of PGP or GPG?
- Do you use a password manager?
- Do you use mobile messaging apps? If so which ones?
- Do you have a background in academia?
- Have you ever heard of Shamir's Secret Sharing?

5.2 Guided walk-through of the features

This guide was never shown to participants, but used as a reference by us as to what functionality we are aiming to test.

5.2.1 Social Backup feature test walk-through

Using this feature requires at least 3 briar users, who take particular roles in the process: One person is the 'secret owner' and the others are all 'custodians'.

The secret owner makes a social backup, looses their phone, gets a new one, and recovers their account with help from their custodians.

Reminder: The aim is to test the social backup feature - not to get general feedback about usability of briar

5.2.1.1 Install package on phones of secret owner and at least two custodians Install the APK of the social backup debug build:

Figure 11: QR code with APK link

http://ameba.ehion.com/download/briar-android-official-debug.apk

Your browser should ask if you want to download the file and when opening it you may have to allow permissions to agree to installing apps from that source.

If you get a message like 'File cannot be opened' without asking about permissions, go to the device settings, search for 'Unknown' and there should be an option called something like 'Install apps from unknown sources'.

Also, with some android versions, Chrome will not let you open the APK, but if you find the downloaded file using the stock file manager app, you can open it from there.

5.2.1.2 Set up briar

- Start Briar
- Choose 'Create new account'
- Choose a name and password
- Connect to local wifi network

5.2.1.3 The secret owner and each custodian add each other as briar contacts On the secret owner, as well as with each custodian:

- On the contacts screen, choose 'add' (the plus sign on bottom right), followed by 'Add nearby'
- Agree to permissions questions
- Scan each other's QR codes.
- Wait for contact to be added. (if this bit doesn't work for some reason, you can also 'add contact from a distance', which involves copying keys and sending them by some other means).

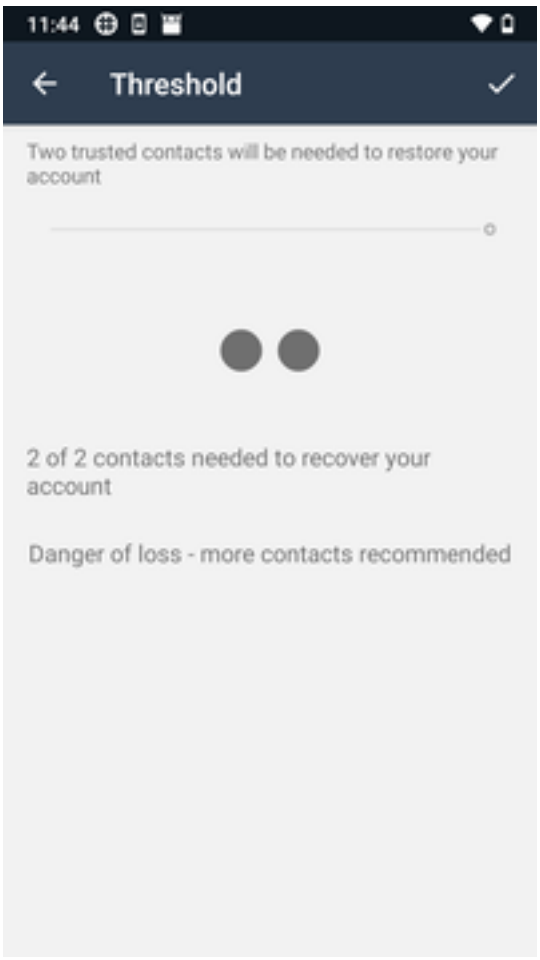

Figure 12: Screenshot of the threshold selector

5.2.1.4 The secret owner makes a social backup Once the secret owner has at least 2 custodians added as contacts, they do the following:

- Open menu (hamburger icon at top left of contacts screen)
- Choose ‘Settings’ and then ‘Social Backup’
- Choose the custodians using the check boxes
- Choose the threshold (if using only 2 or 3 custodians, there will be no choice)
- Click the tick icon in top right corner.

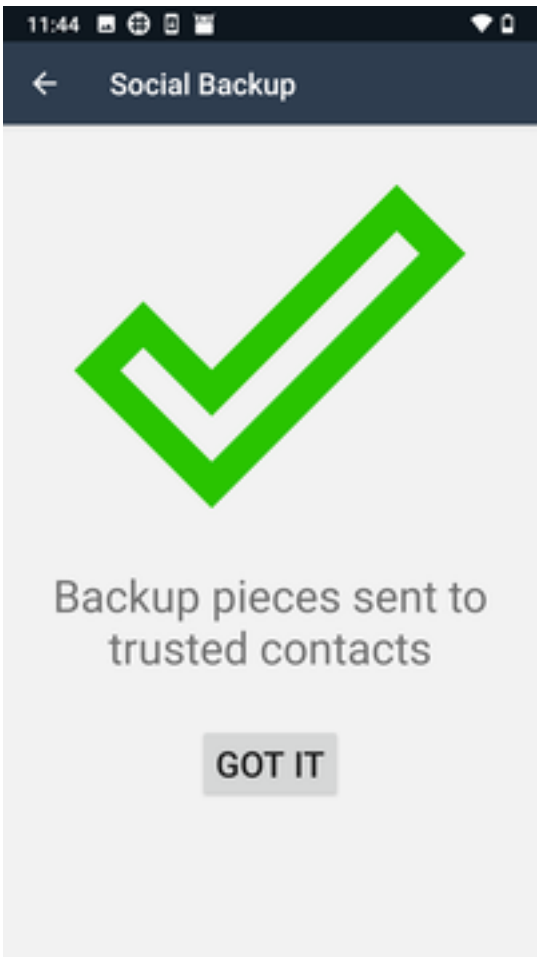

Figure 13: Screenshot of backup pieces sent confirmation

You should see a message to show it worked, and the custodians should get a notification that they have received a message.

5.2.1.5 The secret owner looses their phone Now lets imagine the secret owner looses their phone and gets a new one. We can do this be clearing the app data. The process to do this varies depending on android version, but it is something like: - Close Briar - Open ‘Settings’ - Choose ‘Apps and notifications’ - Choose the briar app ‘Briar SB Debug’ - Choose ‘Storage and cache’ - Choose ‘Clear storage’ - Click ‘Ok’ to confirm

5.2.1.6 The secret owner restores their account Now open Briar again, as though for the first time: - Choose ‘Restore account from backup’ - Click ‘Begin’

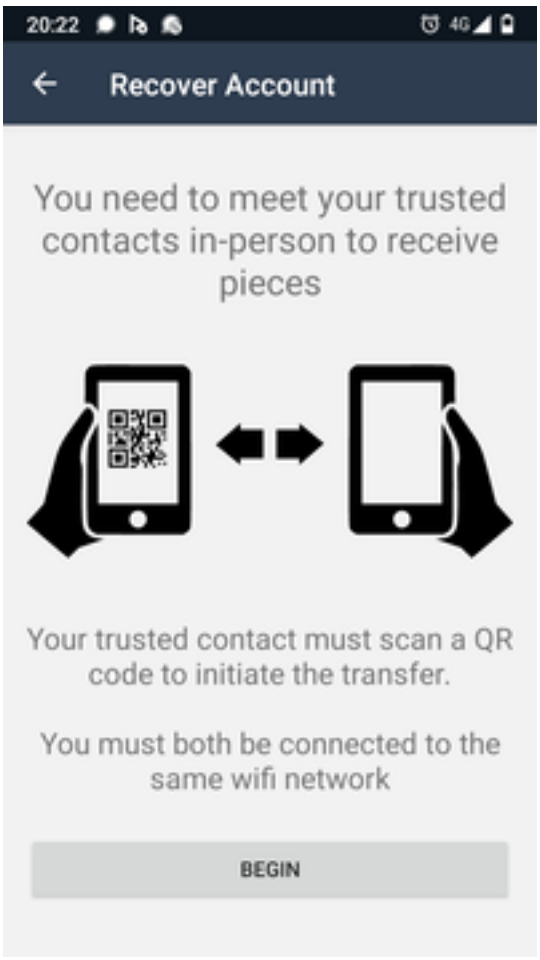

Figure 14: Screenshot explainer screen

One of the custodians does the following: - Choose the secret owner from their contacts list, to open the conversation screen. - Choose the menu icon, with three dots in the top right corner of the screen - Choose ‘Help recover account’ - Click ‘Scan code’

Figure 15: Screenshot showing scanning a QR code

The secret owner clicks ‘Show QR code’ and a QR code is displayed which the custodian must scan.

If the transmission was successful, both secret owner and custodian should see a message to say the backup piece was sent/received. If something went wrong, an error message should be displayed and they can try another time.

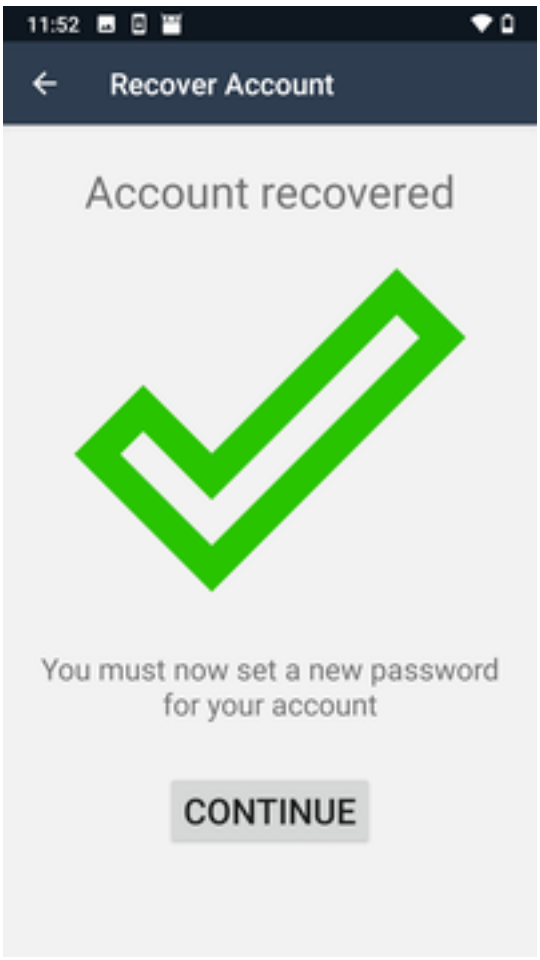

Figure 16: Screenshot showing account recovered confirmation

Once enough backup pieces are recovered, a success message should be displayed, and you will be asked to set a new password.

The account should then be restored along with the original contacts list. You should be taken to the 'Contacts' screen, as when opening briar normally.

5.2.1.7 Reporting bugs If Briar crashes, you should be given the option to send a crash report. Please do send a crash report.

If Briar does not crash but there are some problems, you can send a feedback report which contains logs which will help diagnose the problem. To do this:

- Open the main menu on the contacts screen by tapping the hamburger icon
- Choose 'Settings'

- Choose ‘Send feedback’
- In the ‘enter your feedback field’ - put the words ‘dark crystal’ so we know this is in relation to this feature and not briar generally.
- Tick the ‘Include anonymous data about this device’ checkbox
- Tap the send icon on the top right of the screen

5.2.2 Remote Wipe feature walk-through

Using this feature requires at least 3 Briar users, who take particular roles in the process: - One person is the ‘wipee’ - who can have their account wiped remotely - All the others are all ‘wipers’ - trusted contacts who may remotely activate an account wipe.

Reminder: The aim is to test the remote wipe feature - not to get general feedback about usability of briar

5.2.3 Install package on at least 3 Android devices

Install the APK of the remote wipe debug build:

Figure 17: QR code with APK link

http://ameba.ehion.com/download/briar-android-official-debug.apk

Your browser should ask if you want to download the file and when opening it you may have to allow permissions to agree to installing apps from that source.

If you get a message like ‘File cannot be opened’ without asking about permissions, go to the device settings, search for ‘Unknown’ and there should be an option called something like ‘Install apps from unknown sources’.

Also, with some android versions, Chrome will not let you open the APK, but if you find the downloaded file using the stock file manager app, you can open it from there.

5.2.3.1 Set up briar

- Start Briar
- Choose ‘Create new account’
- Choose a name and password

5.2.3.2 The wipee adds all wipers as briar contacts: On the wipee’s device, as well as with each of the others:

- On the contacts screen, choose ‘add’ (the plus sign on bottom right), followed by ‘Add nearby’
- Agree to permissions questions
- Scan each other’s QR codes.
- Wait for contact to be added.

It will help if both devices are connected to the same wifi network, as otherwise contacts are added using bluetooth which with some devices does not work so well. If it still doesn’t work you can also ‘add contact from a distance’, which involves copying keys and sending them by some other means, eg: SMS.

5.2.3.3 The wipee sets up the remote wipe feature: Once the wipee has at least 2 contacts added, they can do the following:

- Open menu (hamburger icon at top left of contacts screen)
- Choose ‘Settings’ and then ‘Remote Wipe’
- Select the trusted contacts using the check boxes
- Confirm the selection using the tick icon in top right corner.

You should see a confirmation message, and the wiper should get a notification that they have been added as wipers.

5.2.3.4 The wipee gets arrested or captured The wipee’s device is now assumed to be in the hands of an adversary.

5.2.3.5 The wipers activate a remote wipe Two of the wipers do the following: - Choose the wipee from their contacts list, to open the conversation screen. - Choose the menu icon, with three dots in the top right corner of the screen - Choose ‘Activate Remote Wipe’ - An explainer screen should be displayed. Confirm by choosing ‘Activate Remote Wipe’. - A confirmation message should be displayed.

Figure 18: Screenshot of explainer screen

Once two contacts have done this, the wiper's device should be signed out of briar, their account deleted, and Briar should be removed from android's list of recently used apps.

If the wiper is not signed into Briar, or has no connectivity, the wipe will take place whenever connectivity is regained and the messages are received.

The wipe signal messages expire after 24 hours. If the wiper did not receive them after 24 hours, no wipe will occur when connectivity is regained. However, the contacts may re-activate the wipe at any time.

5.2.3.6 Possible questions for participants after the session

- Should the social backup trusted contacts and remote wipe trusted contacts be rolled-into-one? That is, you choose one set of trusted contacts who are both backup-holders and able to activate

a wipe. Are there some situations where you would want to appoint someone to one of these features but not the other?

- Would it be desirable to be able to revoke a contact's ability to activate a remote wipe?
- Is 24 hours a good length of expiration time for wipe messages?
- Two contacts are needed to activate a remote wipe. Is this a good amount? Would it be better to be able to choose the threshold?
- As a person sending the wipe activation signal, would it be useful to get a message confirming that the wipe was carried out? Would such a message pose a security problem?
- Should a message be shown in the UI when signing in following a remote wipe activation? Currently, after logging in, the remote wipe messages are received (providing the device has connectivity and is not in flight-mode), and the account is immediately deleted. What if the user was coerced into entering their password? Could this be a good place to show a message explaining that the wipe was activated remotely and not by the user themselves?

5.3 Explanation of project for user testing session

Before the testing began, we gave participants a short overview of what the project is about:

5.3.1 What is dark crystal?

- A set of protocols, libraries, techniques and guidelines for secure management of sensitive data such as cryptographic keys.
- The idea is to make key management easy by emphasising trust between peers rather than individual responsibility.
- It is a toolkit for developers who want to bring secure and easy to use key management techniques to their project. It is particularly well suited for decentralised systems where authentication relies on keys stored on a peer's device.
- Traditional client-server model - didn't have these key management problems - but at what cost?
- As a test case for our libraries / protocol we are building two features for Briar, a secure messaging app.
- Named dark crystal after the 1982 Jim Henson animated film involving saving the world by putting crystal shards back together

5.3.2 Background of project

Originally funded by Open Technology Fund, currently by National Democratic Institute. Previously received funding from Prototype Fund and Ethereum foundation.

Open Tech Fund - 'committed to advancing global internet freedom', 'supports projects counteracting repressive censorship and surveillance'

- Non-profit and independent - although funded indirectly by United States agency for Global Media
- Support many open source project including Tor and Wireguard VPN

NDI - 'a nonprofit, nonpartisan organization working to support and strengthen democratic institutions worldwide through citizen participation, openness and accountability in government'

- Legal entities in many countries.
- Support a wide range of projects.
- Currently running 'tech summer series' each Wednesday

5.3.3 Briar

Briar is a messaging app aimed at high risk users with security concerns.

It works over the Tor anonymity network as well as other transports. The interface is similar to popular messaging apps, but works in a very different way. When you start Briar, you must set a password.

5.3.4 Social backup

Think about what happens when you loose a device, or it breaks, with the messaging systems you currently use.

There is no server which manages Briar accounts. Your identity is defined by cryptographic keys on your own device, there is no database of all users somewhere, so a Briar account is 'self-authenticated'. Therefore there is no 'send me a new password' button. There is no option to 'send me an SMS to verify it is me on a new device'. Device or password loss means you are locked-out. The feature we are testing is made to fix this problem using 'social backup'.

Social backup is like a 'circle of trust' idea. You assign a 'support group' of trusted contacts who collectively take the authoritative role of agreeing that 'this is me' when you want to recover your account.

Currently the recovery must take place in-person as this is the most secure.

5.3.5 Remote Wipe

Imagine you are an activist and using Briar to communicate with members of a political group. Then you are arrested at a demonstration, and your phone, which is signed into Briar, is taken.

Using the same idea of a support group, you assign people to be able to wipe your Briar account, removing all contacts and messages.