Securing Digital legacies

Josh Muir

22 March 2024

Supervisor: Dr Tristan Henderson
University of St Andrews

Abstract

Digital legacy and what happens to our data after death are becoming a growing concern. This project examines the current state of the art and law surrounding this topic from a privacy perspective and proposes a new solution to pass on digital assets using a digital vault. A digital vault contains data and can be shared with a number of recipients through a threshold secret sharing scheme, where some recipients must cooperate to access the data. The project includes a user study to evaluate the project's success and identify areas for future work. I found that the majority of users found the application easy to use and would use a similar application to manage their digital legacy in the future. This shows the growing need for a legacy solution suited to the digital age that is platform-independent, secure and privacy focused.

Declaration

Contents

1 Introduction

Digital legacy - our digital assets and presence online following death - is becoming an ever greater concern as our lives become more online with our use of online services, digital documents and online discussion forums. It is a broad area, with the term digital assets having various definitions, but in this project, Harbinja's definition - "any electronic asset of personal or economical value" [1] will be used encompassing documents, emails, social media accounts, pictures, bills, digital currency, domains and more.

Given this definition, traditional legacy management approaches do not handle this problem, with a will as a key example. A will cannot contain credentials to access digital assets, as it may become public through the probate process - something that is required if an individual owns any property. It is also a fundamental limitation to how digital assets can be passed on and inherited. Some platform-centric approaches are beginning to emerge (Google Inactive account manager, Apple Legacy contacts and others). While these allow for control over our digital legacy on those platforms, they require extra work for each platform - and with users having an average number 240 online accounts [2] - this approach does not scale. There is an urgent need for a platform-independent, data-focused digital legacy solution.

To consider such a solution, let us first consider a motivating scenario through a collection of personas. Consider Bob, someone who is looking to secure his digital legacy and pass it on to his trusted contacts who are some friends: Alice, Carol, Dan and Frank, and family member Erin. Using data protection terminology, as I will do throughout this project, we can refer to Bob as the data subject and his trusted contacts as data recipients - they will receive his data in the event of his death.

Bob may consider several methods of sharing his legacy currently used in practice, such as sharing all his passwords with data recipients before death or giving each a sealed envelope with instructions to access digital assets [3]. Another method includes going as far to introduce a third party - such as a lawyer - who will only release the data when a certain number of contacts come together (an example given in [4]) These rely heavily on trusting that those contacts and parties will follow the wishes of the deceased - and do 'the right thing' - an unreliable approach given that social relationships and dynamics change over time. By introducing an external technological solution to this problem, Bob can have more certainty that his data will be treated as he wishes and minimise the likelihood of unauthorised access.

This project seeks to provide a mechanism to curate, secure and share important digital legacy assets following the death of a data subject. It does this by maximising user control and privacy and minimising reliance on external parties such as service providers or lawyers. In practical terms, it allows users to import, encrypt and share their data with trusted data recipients through an easy-to-use interface. It provides the ability to override unlock requests - commonly referred to as 'emergency access' in existing tools.

1.1 Objectives

The objectives are this project are divided into primary, secondary and tertiary objectives to determine their importance, and are as follows:

Primary

1. Allow files to be deposited into an encrypted digital vault.
2. Provide a sharing mechanism that allows keys to the digital vault to be shared with trusted individuals, and allow access only when a number of those keys are combined.
3. Support different levels of trust for different individuals.
4. Make the system compliant with relevant privacy law.

Secondary

1. Use distributed systems or other technology to improve the resilience of the system.
2. Support cancellation functionality, so that vault owners can cancel any unlocking of the vault if they are still capable.
3. Build a system to keep the vault up to date over time.

Tertiary

1. Interoperate with existing solutions, for example Facebook's memorialising accounts or Google's inactive account manager.
2. Use media and techniques to ensure the system will be capable of lasting a human lifespan.

2 Context survey

This section will examine the motivations for this project, and the current state of technology and the law, as relating to digital legacy.

2.1 State of the art: What platforms are doing now

Technology companies have recently caught onto the need for ways of managing their users digital legacy, and what happens to their personal data when they die. This is clearly necessary, as it has been predicted that the number of profiles belonging to deceased individuals could be more than living before the end of the century [5].

Examples of the state of the art include Google's Inactive account manager, Apple Legacy contacts and Facebook memorialised profiles. These tools can provide fine-grained control over accounts and data, but they suffer from the fundamental limitation: They are platform-specific. That means data subjects must take the time to individually set up legacy controls on each.

2.1.1 Google inactive account manager

The inactive account manager [6] provides the most comprehensive controls of the platforms surveyed, and is triggered automatically when a data subject does not engage with the service for a set amount of time. Data subjects can set it up to alert up to 10 people when they do not use the service, and can customise the data to be shared with each data recipient. After a pre-determined further period, the data is then automatically deleted. This represents a comprehensive way of managing personal data, based purely upon service usage, with no need for death certificates or other documentation. This works well for Google accounts, which tend to be linked to mobile devices and regularly used.

Google's approach also focuses less on the concept of digital legacy, and data management following death, and instead focuses upon whether the service and stored data is in active use. It is very possible that a subject's personal data could be put through this process simply because it is no longer used—rather than the death of the data subject.

2.1.2 Apple legacy contacts

The legacy contact [7] offering from Apple is similar to the concept of a digital vault: Digital legacy contacts can be created, and each is given a key. By presenting this key and a death certificate to Apple, it is possible to receive access to the data subject′s iCloud account—but not sensitive data such as passwords. This access is time-limited, and after 3-years the data is deleted [8]. This represents an effective solution for managing the iCloud data—which is especially important given for Apple customers, a great number of their photos, memories and correspondence may be stored in the account. The decision to include passwords in this data access means that this solution alone cannot allow for access to the data subject′s wider digital legacy - their accounts, profiles and assets on online services. Given that aspect of digital legacy that users are most concerned about is their photos [3], this approach may be an acceptable balance between passing on data and maintaining the privacy of the individual, without requiring further configuration. Even without passing on passwords Apple requires some preparation before death - if legacy contacts are not set up, or if you have a death certificate but are not a legacy contact, Apple permits account deletion but not access [7].

2.1.3 Facebook memorialised accounts

Facebook, meanwhile, has an account memorialisation feature [9] which means by default accounts are put into a read-only state indicating that the profile owner has died. Data subjects can assign data recipients by creating legacy contacts. In a similar vein to Apple and Google, data recipients can manage the account in a limited way, posting a status update, moderating posts and requesting account deletion as necessary. Similar to the Apple approach a death certificate is required to memorialise an account - or a copy of a will and obituary.

One concerning aspect of Facebook's approach is that by default accounts are memorialised, meaning if no next-of-kin exists, the profile will exist read-only in perpetuity, providing potential for the harvesting of personal data to create Ghostbots [10] and related harms.

2.2 The law: How should this information be governed?

The law surrounding digital legacy and our right (or lack thereof) to privacy after we die is currently a topic of much discussion.

In the UK, under data protection legislation (most recently the UK GDPR as implemented by the data Protection Act 2018 [11]), privacy is considered to be a personal right - meaning it dies with the individual, and others cannot assert privacy rights on the behalf of a data subject [12]. The UK GDPR affords significant control to data subjects (that is, those identified by the data in question) and gives them rights such as the right to be informed, of access, of rectification, of erasure, to object and others [13] as well as setting out broad principles that apply to processing.

It is notable that while data protection legislation has largely kept pace with advances in technology, the law surrounding digital legacy has not. The sheer volume of data about individuals being gathered and curated - and the range of digital assets created by those individuals themselves is far greater than equivalents in offline media, and there is little to govern how those assets and data are treated after death. Traditional succession offers little help here given data is not property so does not form part of an estate, and so does not pass on through traditional succession means [14].

One possible solution to this, proposed by Harbinja, is to introduce the notion of post-mortem privacy which she defines as "the right of a person to preserve and control what becomes of his or her reputation, dignity, integrity, secrets, or memory after death." [15] and allow users, through contracts or configuration technology such as those described in Section 2.1 to set their wishes. She also argues that in the absence of any clear wish the data should be deleted by default [14] on the basis that those digital assets are inherently sensitive and private information that should only be disclosed according to the instructions of the data subject.

Privacy only being a right of the living is in apparent contrast to the duty of medical confidentiality - which seems at first to continue beyond the death of the individual. However, medical confidentiality is only maintained beyond death to maintain the privacy and lives of the family and medical personnel involved with the deceased [16]. This same change in focus to the living can be seen following a major disaster, or accident. The taking and releasing of photographs or videos of the deceased is met with widespread condemnation - as it is not respecting the families right to privacy and right to grieve [17]. This concept of "familial/survivor privacy" is recognised by the courts as a common law right in America [17].

This same approach could be a way forward for post-mortem privacy: A change to focus on the living and family members, and like with physical assets, a focus on letting them manage the digital property - and data - of the deceased.

In the US there have been recent developments in the laws surrounding digital legacy. The Uniform Law Commission has drafted the Uniform Fiduciary Access to Digital Assets Act. Controversially, the law had a focus on the passing digital assets to next of kin but ignored the privacy aspects—and mostly as a result of this failed to be passed in its original form in many states [12, p17].

It becomes clear, then, that the right of heirs to access digital assets must be balanced against some notion of post-mortem privacy and that we can most easily resolve this by documenting the way we want our data to be shared. A digital vault could fulfil this aim—providing explicit access to selected information to selected individuals in a way backed by cryptography - and leaving the rest to be deleted.

2.3 Last will and testament: Why a digital vault?

A digital vault can bridge the gap between platforms and their digital legacy implementations, and provide fine-tuned control over who gets access to data about a data subject, without relying on an external authority to adjudicate on whether that access should be permitted.

It may be possible to integrate with these platforms, using their third party Application Programming Interfaces (APIs), to carry out the wishes contained within a digital vault automatically, when the vault is opened. This would effectively provide a uniform way of managing digital legacies — or at the very least gathering in one place the wishes of the data subject regarding these digital assets and online accounts.

If platforms do not add such third-party support, it is still possible to share account login details with recipients, but the sharing of account login information is another way traditional wills fall short. With the possibility of a will becoming a public document if a grant of probate is issued [18, 2.1], passwords would be left accessible to all—not an effective sharing strategy.

With a digital vault, these details can be securely shared and be made accessible only to those who should have access. This does come with a significant drawback that merits further study—most Terms of Service for online platforms prohibit transferring ownership of online accounts. This means that if the company were to become aware of the death of the individual, these authentication details could be disabled. This is an area that legislation could provide more uniform access or a framework for handling such access.

Digital vaults, in a conceptual sense, are essentially an implementation of secret sharing. Secret Sharing is used in many fields, including in the control of nuclear command and control systems [19]. The scheme used in this project was originally introduced by Shamir [20] in 1979, and refers to a threshold (k,n) threshold scheme where having k or more pieces of a secret means it can be computed, and any less it cannot be computed - and that those pieces do not reveal any information about the secret. Shamir sets out this scheme in the context of sharing a cryptographic key and that is an important consideration. Each piece of a secret shared in this way is at least the same size as the secret and splitting the secret directly would make it difficult to modify or update it at a later date. In this context we can split a key into many key pieces which must then be combined to get access to the original key, and thus the encrypted data.

The ability to digitally require the cooperation of several people through this scheme is a very useful property especially in the context of a digital vault, as it can mitigate against rogue data recipients. Under the Apple or Facebook schemes, for example, a recipient could present falsified documentation to the company to gain access to the person's account for malicious purposes. Using a secret sharing scheme protects against this, raising the bar for such an attack, as multiple recipients would need to collude in order to get access to the data.

It is reasonable to assume, given the increasing digitisation of our lives and as legal processes become more online that a digital last will and testament will become normal. This opens opportunities for new ways of sharing wishes and providing closure to next-of-kin, for example, with a digital vault, a video introduction could be included to explain its contents and be of significant sentimental value. It makes it possible to deliver the wishes of the deceased automatically, for example by defaulting to deletion and disclosure to the next of kin [3].

In this project I aim to fill the gap left by platform-specific digital legacy solutions and provide a way of passing on personal data and sharing the wishes of data subjects surrounding their legacy. The project will do this without depending on external "trusted parties‘ (such as lawyers) or trusted documents such as death certificates as is traditionally required for succession.

3 Requirements specification

Given the context described in the previous section and objectives of the project, we can set out requirements for a system that will provide a unified digital legacy solution with a focus on user control and privacy. I will continue to use the personas described previously: Bob (data subject), Alice, Carol, Dan, Frank and Erin (data recipients).

3.1 Functional requirements

1. The system must allow Bob to specify how many of his data recipients are required to decrypt his personal data.
2. The system must allow Bob to specify different levels of trust for different data recipients: for example family and friends.
3. Bob must be able to update the data without the key pieces.
4. Data recipients must be able to verify that Bob created the encrypted data and key pieces.
5. Data recipients must be able to access Bob's vault and the data in it given the specified number of key pieces, and that Bob is no longer alive.
6. The system should interoperate with Bob's social media accounts.

3.2 Non-functional requirements

7. It should be mathematically impossible to unlock the personal data unless Bob is no longer alive.
8. The system must be usable by a non-technical person, and require no understanding of the underlying cryptography.
9. The system should be able to decrypt the personal data with no internet access.
10. It should be easy for Bob to keep his stored personal data up to date.

3.3 Domain requirements

11. The system should be capable of surviving the average lifetime of a human in terms of medium and software compatibility.

Table 1: Evaluated language & tool criteria table

Criterion	Electron & Node	Rust (Tauri/React)	Java	Flutter	C
Mature	✓	×	✓	×	✓
Maintained	✓	✓	✓	✓	✓
Can make UI	✓	✓	✓	✓	✓
Cross-platform	✓	✓	✓	✓	×
Supports OpenSSL	✓	*	✓	×	✓
HTTP Server support	✓	✓	✓	×	×
AWS Library	✓	✓	✓	×	×

4 Software Engineering Process

The software for this project was developed using an incremental, Agile approach. Much of the initial work was focused on design and an in-depth understanding of the context, and how the problem of digital legacy could be solved in a way that interoperated across providers and data sources.

Implementation was split into one-week sprints, coinciding with supervisor meetings. The first two sprints were longer and focused on speed to implement the basic functionality of the system, before moving to a more stable sprint structure. The first focused on design - both of the application user interface and how the creation process could be made user friendly - and on the architecture and way user data flowed through the system.

Once using a more structured sprint approach, tasks to be completed in the following week tasks completed prior were maintained using weekly progress reports. This, combined with a kanban board in Github projects, provided both an overview of overall work to be done and the progress in a given week. The kanban board contained user stories and each contained included specific, precise acceptance criteria which indicated what had to be implemented for the story to be 'complete' and merged to the main branch of the project.

Objectives for each week were set in a meeting with my supervisor, and appropriate stories moved into the backlog for that week.

4.1 Technology

Using the architecture design and requirements for this project, a list of criteria can be generated to evaluate potential technology choices. Table 1 shows some technologies, and how they meet the criteria. The symbols used are × to indicate supporting an unsupported criterion, * to indicate partial support and ✓ to indicate full support.

Partial support means, for example, that a library exists but it is marked as in testing or not ready for production use. Based on these criteria I originally chose to create the desktop application with Electron and the server backend with Node.js. This has the advantage of being fully cross-platform, presenting the opportunity to turn it into a website in future and I am already familiar with it.

Prior to implementation I changed this decision to use Rust, which involved using a tool called Tauri [21] to create user interfaces with TypeScript, and then do the underlying work with Rust. I made this decision as the Rust AWS library became stable after the original decision was made, and Tauri allows the application to consume much less resources and be fully cross platform - which would help users if they have an older device. One of the criteria, the presence of an OpenSSL library or binding was not as relevant as expected as secure, audited cryptographic Rust code existed which was not part of OpenSSL.

5 Ethics

This project was undertaken under the standard software artifact form, which is attached as appendix A.2.2. Synthetic data was used for all testing and user study of the program, as the use of real private or sensitive data or will information would not be appropriate.

6 Design

This section will explore the design of the system from an architectural, security and privacy perspective before looking at the design of the interface itself.

6.1 Architecture

The architecture of the system as a whole is important to support user privacy and control over their data and to fulfil requirements 5, 7 and 10: It should be impossible to access the data unless Bob is dead or consents to the access; his recipients can do so given the required key pieces, and Bob can easily keep his data up to date. To support these requirements, I created a desktop application supported by an optional cloud provider - which would provide data safety and security assurances as discussed in Section 6.2. For this reason, I focused on the personal data, its sources, and how it flows through the whole system structure - shown in Figure 1. The architecture is designed according to requirements 5, 7 and 10.

The flow is as follows:

1. The data subject, in this case Bob, puts their personal data into the application.
2. Bob configures the application so that it uses their desired vault type, generates the necessary number of keys and uses extra features (like unlock notifications) if desired.
3. The vault is encrypted and either uploaded, provided to Bob or both.
4. Bob will then distribute the generated key pieces to his data recipients.

Then following the death of Bob, his data recipients can come together and combine their key pieces on one computer using the application and be provided with the data. Note that for an offline vault, the data recipients would also need to provide a copy of the vault file, and for a cloud-backed vault an unlock notification may have been applied. The diagram shows recipient access through both an offline file and cloud vault, but only one is required. This dual design according to the requirements 7 and 9 and due to the trade-off between the two: It is not possible to create a system that both ensures Bob is deceased and allows for offline unlocking (as there is no way for to determine his status in such an environment), so the best solution is to give the user the choice of which to use.

The system can be broadly separated into three locations: The computer of the data subject (Bob in this example), the cloud infrastructure and the computer of the data recipient performing the unlock operation. Data and configuration are inputted into the application, which encrypts it and generates a main key and an unlock key. The encrypted data is then provided either as a file to the user (an offline backup) or uploaded to the cloud gateway.

This architecture maximises user privacy, security and control - the cloud service only has access to the encrypted data (a form of zero knowledge storage - the server

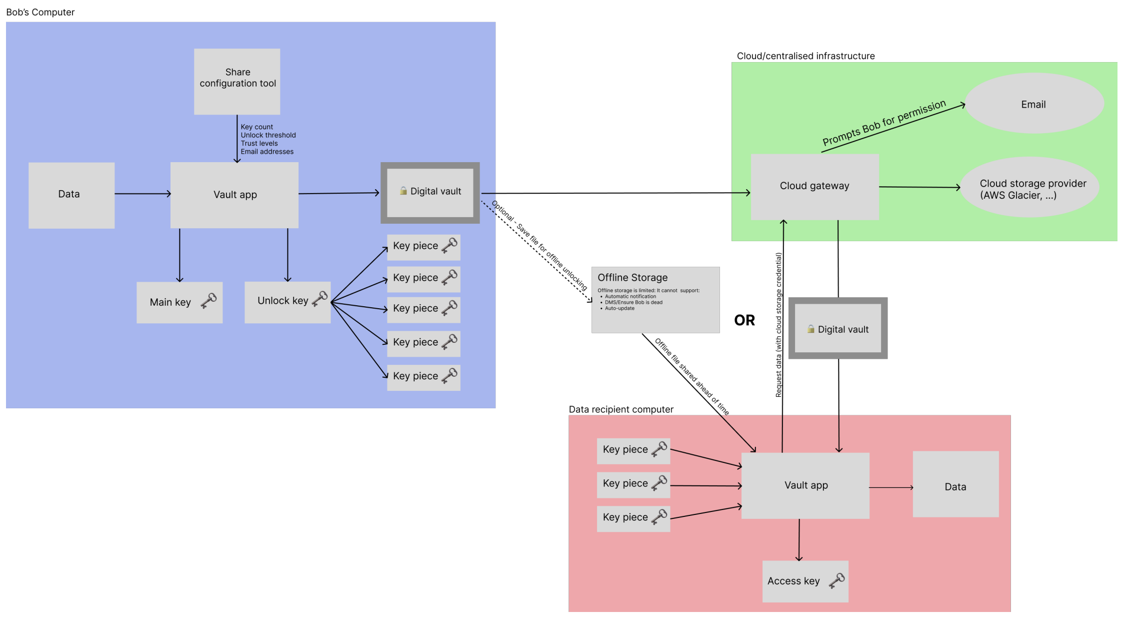


Figure 1: Architecture diagram showing data flow through the system and the different ways a recipient can access the data.

holds the encrypted data but not the keys to decrypt it), and they have the option to use the offline option instead of - or in addition to - the cloud service. In an ideal world, the cloud service would not be required. However, it is the only way to implement time-bound decryption and an unlock notification. An unlock notification is required to meet requirement 7 and prevent data recipients from colluding to combine their keys and accessing the data without Bob's consent or not at the appropriate time. This functionality means that for a cloud-backed vault when an unlock attempt is made the recipients will need to wait for the notification period, ensuring that Bob is either dead or consents to the attempt.

It also supports requirement 5 that given the required key pieces, the recipients can access the data about Bob, as with a cloud-backed vault they do not also need to get a copy of the vault file.

6.2 Attack tree & Security model

Given the application deals with important user data and that there is potential for malicious attempts to access the data - either by recipients or third parties - it is important to consider the threats and set out an appropriate threat model.

Figure 2 is an attack tree which sets out the possible attacks against the system and vault data. Initial threat modeling showed potential for confidentiality of the personal data to be compromised - especially by colluding recipients or compromised key pieces. To protect against these attacks and to meet requirement 7 I implemented unlock notifications, which are discussed further in Section 6.2.1. The attack tree shows that the

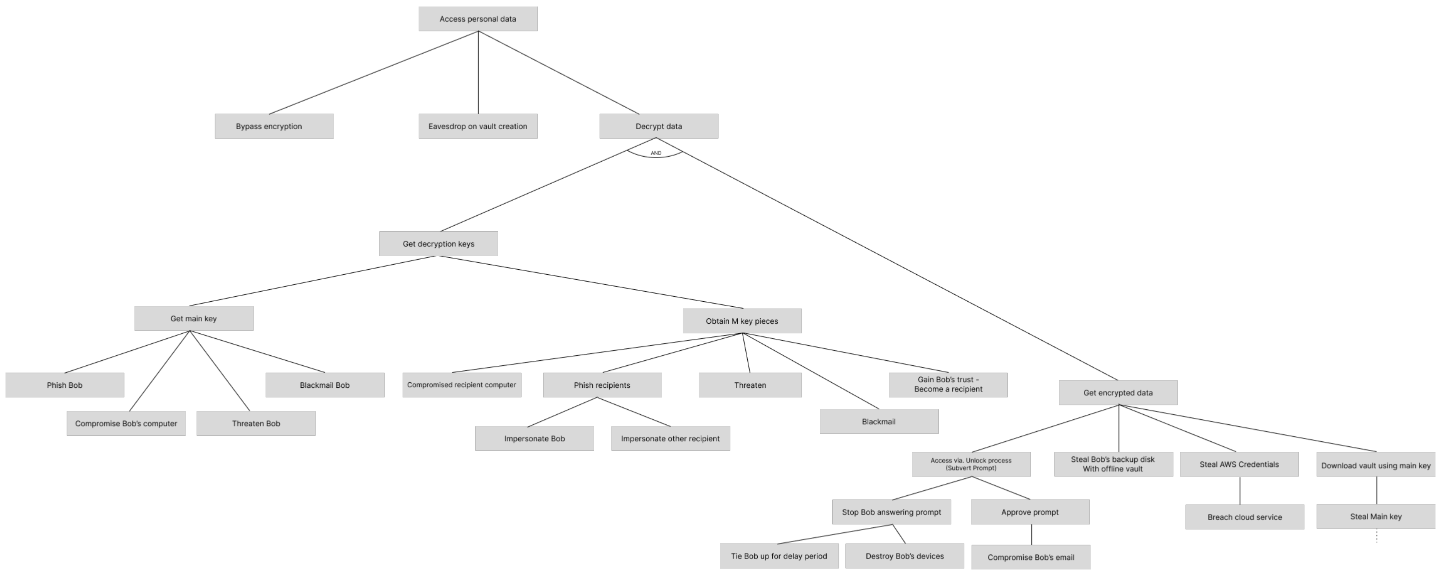


Figure 2: Attack-Defence tree showing unauthorised data access. Note that the level of boxes is not significant, as re-arrangement was nessessary to make the figure readable.

inclusion of an unlock notification significantly increases the difficulty of attacks against user data, as the server itself must be compromised - or the data subject must be physically prevented from denying the request. The other possible vulnerabilities, including eavesdropping or subverting the encryption algorithm, are accepted and considered outside the scope of this project and mitigated by using widely trusted encryption algorithm implementations.

Assuming the server remains secure, there remains two main avenues for attack: Attackers must get both a set of keys and a copy of the encrypted vault file, and there are a number of ways to do this. With a cloud-backed vault, if the attacker gets a copy of either the required key pieces or the main key, they can download a copy of the vault. This is subject to the unlock notification grace period if using the key piece unlock mechanism, but there is no additional protection for the main key, so it must be kept securely.

I considered adding a password to the main key (similar to PGP or SSH keys) where the key is stored encrypted on disk, but felt that the barrier to usability for this feature would be too high. This could form part of future work to integrate a password option, or use a secret store such as a Trusted Platform Module.

6.2.1 Unlock notification "Grace period"

The unlock notification functionality means that if a data subject is still alive, their data cannot be accessed - even if the correct number of keys are combined. I implemented it to meet requirement 7. If enabled, unlock notifications mean it is mathematically impossible to access the data unless Bob consents or is deceased.

It works by taking a user preference for unlock time - for example, 7 days - and only releasing the encrypted vault data from the cloud provider after this time. To allow the data subject to prevent the unlock from taking place, they will receive an email with allow and deny links, and if they respond during that period, they can reject the request.

The disadvantage of this approach, as discussed above, is that it adds an external dependency (the cloud service), but it also provides resilience to data loss of the original file. I also considered an alternative design, where the cloud provider held a key piece instead of a copy of the vault itself but chose to upload the encrypted data. Uploading a key piece would yield the same disadvantages without providing added resilience to data loss.

Other approaches include a 'dead man's switch', which would regularly email the data subject to confirm they are still alive. This is similar to the approach used by Google's Inactive Account manager [6]. However, the unlock notification is a better approach for the digital vault as it does not require regular, meaningless action to protect their personal data. It is also widely used in similar applications such as password managers [22–25] and 7 days is the most common delay time.

6.2.2 Sharing & key configuration

Looking at requirements 1 and 2 we need to consider a secret sharing scheme. It can be seen in Figure 1 that the data subject receives a "main key" which allows them to update and access the data, while data recipients receive a key piece. Having separate data subject and data recipient keys means Bob can keep his data up to date without combining key pieces or bringing his recipients together - as per requirement 3. The design for the sharing component was first based upon a single tier of trust, where every key piece holder is trusted equally.

This is the simplest form of threshold scheme originally envisaged by Shamir and can be implemented in a very simple of way by specifying two numbers - the number of keys required and the number of total keys often referred to as a k of n scheme. Using such a scheme the secret that was split can be accessed by combining any k of the n pieces but having up to and including $k-1$ pieces provides no information about the secret. For this project, the secret that is being shared is a key that can be used to decrypt the data - so I refer to the pieces as key pieces.

While simple, this method ignores much of the complexity of social circles and trusted contacts, and could result in situations where all the least trusted individuals combine their keys and access the data. To solve this, a scheme with two levels of trust was devised and implemented as described in section 7.1.1.

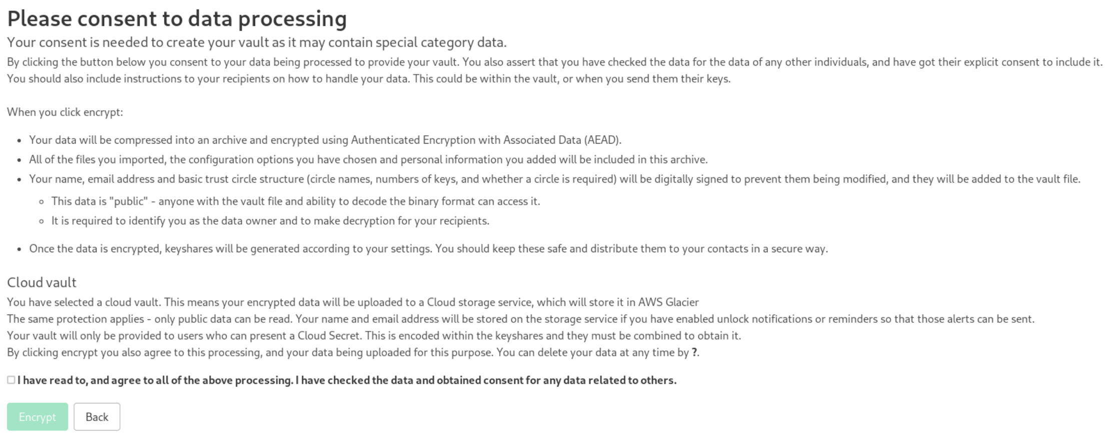
6.3 Privacy & Legal compliance - Data Protection Impact Assessment

A Data protection impact assessment (Fig 3) was carried out to assess the privacy implications of this project as a piece of new technology and to comply with Primary Objective 4. It identified some potential issues, particularly around the inclusion of data identifying other individuals by the data subject, and it is included as Appendix A.4.

"A DPIA is a process designed to help you systematically analyse, identify and minimise the data protection risks of a project or plan."

Figure 3: ICO Website - What is a DPIA? [26]

Following the DPIA process, I made two changes to the design of the system: First, to ensure that the data subject is fully aware of what they are doing by encrypting the vault, I added a consent page to the design, and it only lets the user progress past it once they agree to it. This is especially relevant as the vault may contain special category data (if the owner chooses to include it) - so explicit, informed consent is essential. The final consent page is shown in Figure 4.



Please consent to data processing

Your consent is needed to create your vault as it may contain special category data.

By clicking the button below you consent to your data being processed to provide your vault. You also assert that you have checked the data for the data of any other individuals, and have got their explicit consent to include it.
You should also include instructions to your recipients on how to handle your data. This could be within the vault, or when you send them their keys.

When you click encrypt:

- Your data will be compressed into an archive and encrypted using Authenticated Encryption with Associated Data (AEAD).
- All of the files you imported, the configuration options you have chosen and personal information you added will be included in this archive.
- Your name, email address and basic trust circle structure (circle names, numbers of keys, and whether a circle is required) will be digitally signed to prevent them being modified, and they will be added to the vault file.
  - This data is "public" - anyone with the vault file and ability to decode the binary format can access it.
  - It is required to identify you as the data owner and to make decryption for your recipients.
- Once the data is encrypted, keyshares will be generated according to your settings. You should keep these safe and distribute them to your contacts in a secure way.

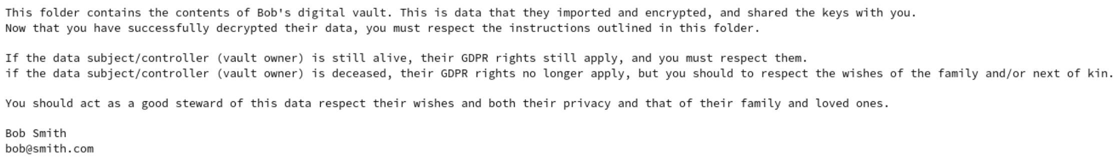
Cloud vault

You have selected a cloud vault. This means your encrypted data will be uploaded to a Cloud storage service, which will store it in AWS Glacier
The same protection applies - only public data can be read. Your name and email address will be stored on the storage service if you have enabled unlock notifications or reminders so that those alerts can be sent.
Your vault will only be provided to users who can present a Cloud Secret. This is encoded within the keyshares and they must be combined to obtain it.
By clicking encrypt you also agree to this processing, and your data being uploaded for this purpose. You can delete your data at any time by **?**.

☐ **I have read to, and agree to all of the above processing. I have checked the data and obtained consent for any data related to others.**

Figure 4: Screenshot showing the consent page, which is shown just before data is encrypted

Second, To inform data recipients and encourage them to use the data appropriately, a privacy notice is added to all the vault archives so when data recipients decrypt the data they are aware of their obligations. Figure 5 shows one of those consent pages.



```
This folder contains the contents of Bob's digital vault. This is data that they imported and encrypted, and shared the keys with you.
Now that you have successfully decrypted their data, you must respect the instructions outlined in this folder.

If the data subject/controller (vault owner) is still alive, their GDPR rights still apply, and you must respect them.
if the data subject/controller (vault owner) is deceased, their GDPR rights no longer apply, but you should to respect the wishes of the family and/or next of kin.

You should act as a good steward of this data respect their wishes and both their privacy and that of their family and loved ones.

Bob Smith
bob@smith.com
```

Figure 5: Screenshot showing an example privacy notice. These notices are included in the extracted files as privacy_notice.txt.

6.4 User Interface Design

To meet requirement 8, I designed a user interface focusing on ease of use so that non-technical users could use it and allow them to carry out the core actions - create a vault, open a vault and unlock a vault given the key pieces.

I completed a user interface design, using a story board technique - incrementally designing each page of the user interface. The design was created using the Figma design tool [27]. When creating the interface, I used the Nielsen usability heuristics [28] as a reference, and the application was split into multiple pages as recommended by the gov.uk style guide [29], with each page focusing on a single concern. I chose to follow the gov.uk guidance as the government websites contain a wide range of complex forms designed for use by a wide range of users and are highly rated for their accessibility. The original interface designs also had the 'Continue' button floating on the right side of the screen, which was changed to be left aligned and always on the outer edge to match user expectations [30, 31].

Some key pages of the interface design are shown in Figures 6, 7 and 8 and the appearance of the final user interface is discussed further in the implementation section.

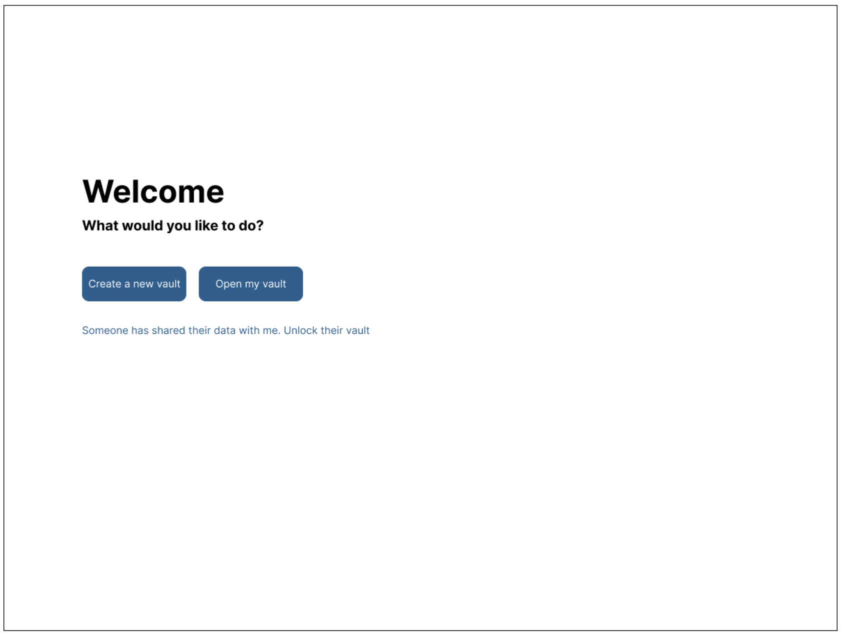

Welcome
What would you like to do?
Create a new vault
Open my vault
Someone has shared their data with me. Unlock their vault

Figure 6: Wireframe showing the welcome screen.

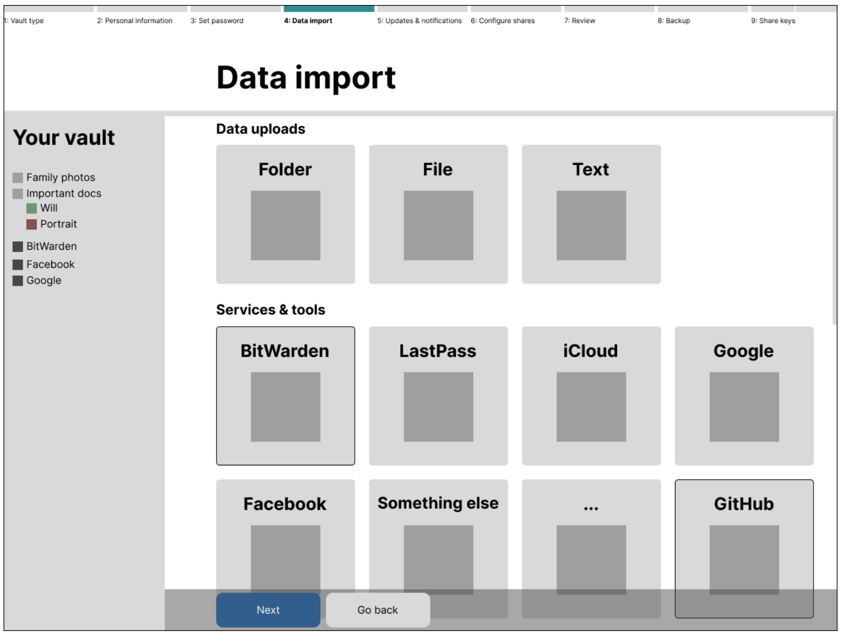


Figure 7: Early design showing a possible file import screen. This design was not used in the final application.

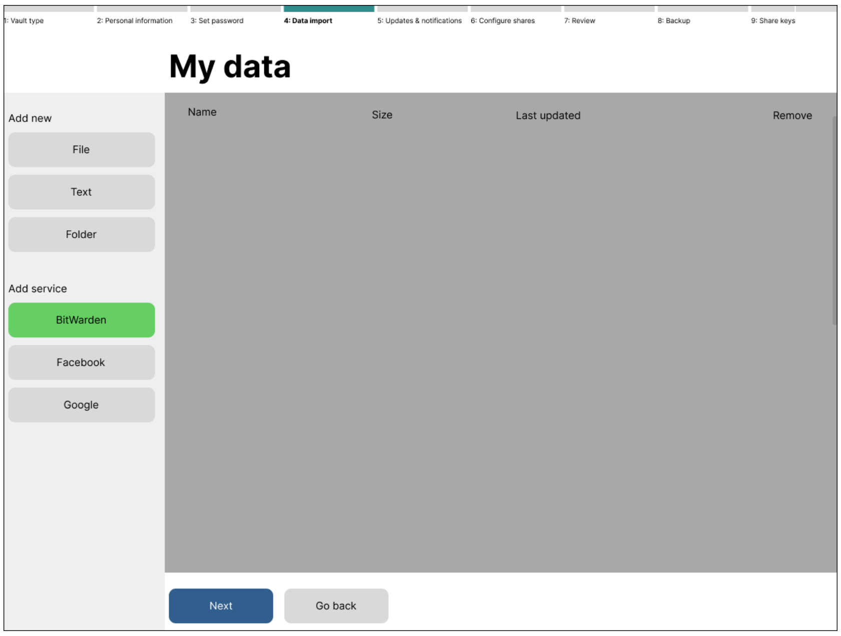


Figure 8: Updated design showing import buttons on the left, and a file table on the right.

7 Implementation

In line with the iterative development approach described in Section 4 I first focused on a subset of the requirements before looking to implement others. The first requirements I focused on were requirements 1, 2, 3, 4, 5, 8 and 9.

7.1 Desktop Application

The Desktop application was implemented using Tauri with React & Typescript as chosen in Section 4.1. This means there was a display/user interface layer written in Typescript & React, and a 'backend' lower layer written in Rust. The upper layer handles validation, displaying errors, pages, http requests and some basic file operations while the bulk of cryptography and vault management is performed in the Rust layer.

To create the user interface (seen in Figure 10), I also used the Bulma CSS library [32] for some layout and styling. I created the user interface according to the UI design, focusing on a series of pages that the user can move through. Each of the main actions the user can perform has its own page flow (where a page flow is a sequence of pages for a given action) and those flows are vault creation, vault opening (using the main key) and vault unlocking. With each page focused on a single concern with some only conveying information (for example the consent page) while others allow the user to configure some related attributes (circles, for example). It would not be appropriate to separate the text description on what a circle is, and how to configure one from the share configuration page as this would be a violation of Nielsen's sixth heuristic [28] - recognition rather than recall - and require the user to remember a lot of information.

The page concept was very useful for displaying a progress bar and showing the user their progress through the creation process as it is fairly long. The user vault creation flow is as follows:

- Select vault type - The user chooses whether they want a cloud-backed or offline only vault. An offline copy can still be downloaded later.
- Personal information - The user inputs their name and email address, and optionally extra information such as legal name and phone number that may be useful for their recipients.
- The user imports their data - files and folders from the file system.
- If it is a cloud-backed vault, the user can then configure alert notifications and update reminders.
- Next, the user sets up sharing using circles and key pieces.
- Next the user is asked to review their inputs and consent to processing before the encryption takes place.
- And finally the user can save a copy of the vault file and export their keys.

The processes to open and unlock vaults are largely the same. For an offline vault, a file must be provided and then either a set of key pieces or the main key depending on the flow chosen. For a cloud-backed vault meanwhile, the keys are submitted first and then the vault is downloaded and decrypted - subject to alert notifications for the unlock flow.

Figures 10 and 11 show some of the interface pages for the application.

One notable design feature on the Updates &Notifications page shown in 11 is the unlock time input box. I made the box support various time units (hour, day, week, month) to allow the user to fine-tune their presences for alert duration and their selection is converted into seconds to be used elsewhere in the application. This is an improvement compared to many existing unlock notification implementations which usually allow a fixed delay of 7 days.

A vault can be updated by opening it and then editing its contents. For simplicity, I chose to make it so that circles cannot be updated. While it is technically possible to update shares (i.e. add more for the same configuration) I felt this may add confusion for the user as they may believe previous keys have been revoked. For this reason the only way to revoke existing keys is to delete the vault and create a new one. Figure 9 shows the share configuration page with fields disabled.

Another notable design feature is the file browser which supports add files and folders and nesting folders as with a real file browser. I implemented it as an interface onto the file system rather than re-implementing file system functionality by creating a file tree. It creates a temporary folder within the program directory and files and folders are added to this temporary folder as they are added in the interface, and as the user navigates between folders the actual contents of the vault folder are returned. This method provided an easy way of producing a bug-free implementation of nested folders that is robust, and it has the advantage that all the data is preloaded and ready for the encryption stage.

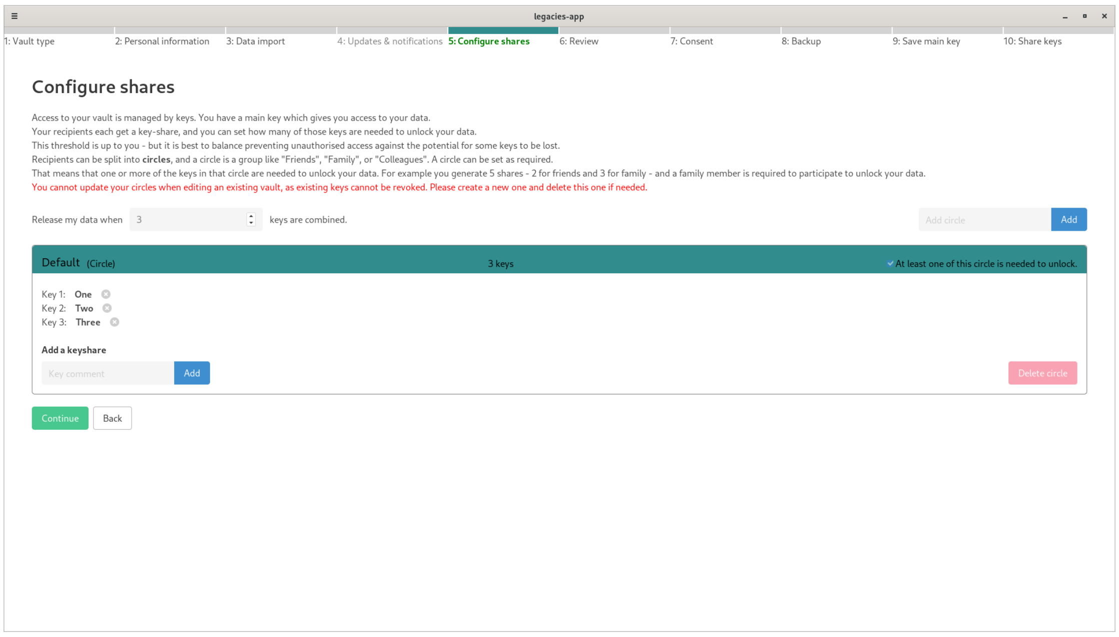
7.1.1 Cryptography

To implement the scheme described in Section 6.2.2 I used Authenticated Cryptography (chacha20poly1305 [33]) to encrypt the vault and sign the metadata attached to it, and this uses a 32-byte key. To produce a scheme with multiple levels of trust, I used the technique described in [34] and Figure 12 shows the pseudocode I created for my implementation.

As shown in Figure 12, The 32 byte key is then split linearly into M+1 pieces, where M is the number of 'required' circles. Each of those linear keys is allocated to a required circle, and the final one is split into N pieces. Each member of a required circle is given their circle key piece and their individual key piece, while other recipients are given only an individual key piece.

This means that the circle key-share of all required circles, plus the required number of individual keys must be combined to get the original key back.

I chose to import the core encryption and secret splitting algorithms from existing Rust creates (libraries) instead of re-implementing the algorithm according to the



Configure shares

Access to your vault is managed by keys. You have a main key which gives you access to your data.
Your recipients each get a key-share, and you can set how many of those keys are needed to unlock your data.
This threshold is up to you - but it is best to balance preventing unauthorised access against the potential for some keys to be lost.
Recipients can be split into **circles**, and a circle is a group like "Friends", "Family", or "Colleagues". A circle can be set as required.
That means that one or more of the keys in that circle are needed to unlock your data. For example you generate 5 shares - 2 for friends and 3 for family - and a family member is required to participate to unlock your data.
You cannot update your circles when editing an existing vault, as existing keys cannot be revoked. Please create a new one and delete this one if needed.

Release my data when 3 keys are combined.

Default (Circle) 3 keys ✓ At least one of this circle is needed to unlock.

Key 1: **One**
Key 2: **Two**
Key 3: **Three**

Add a keyshare

Figure 9: Screenshot showing the share configuration page for an existing vault

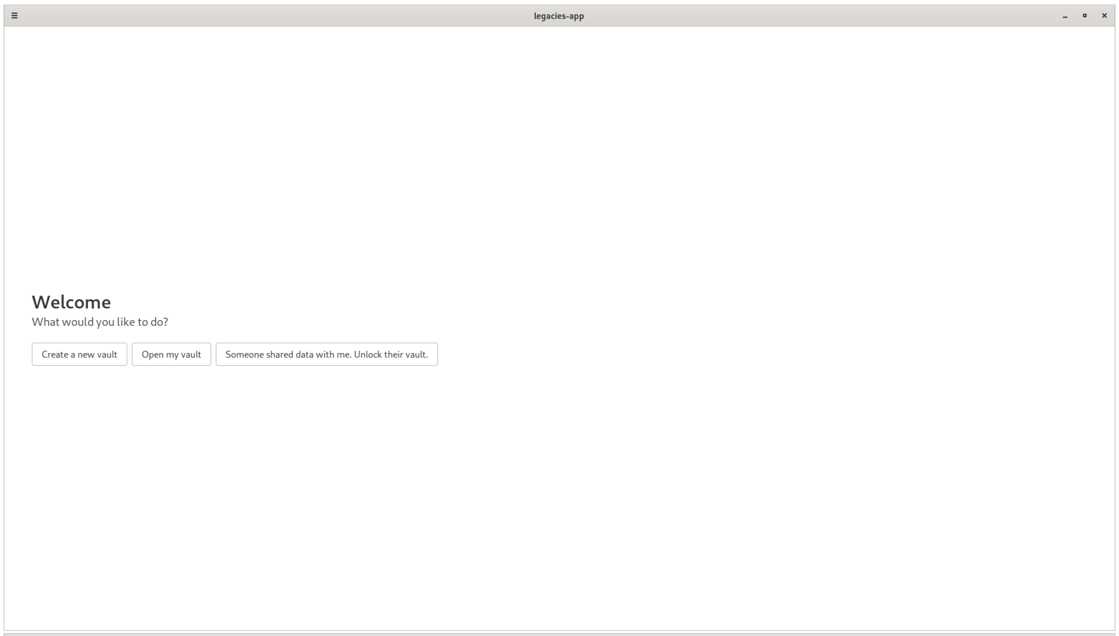
principle of “Don’t roll your own crypto” commonly attributed to Bruce Schneier [35]. This is because any implementation I could create would likely be flawed - and this way the project relies on widely used and verified implementations. These algorithms are then used according to the schemes described in Figures 12 and 13 to actually provide multiple levels of trust.

The decryption process for keys is much simpler than the encryption, and is described in Figure 13. Sets are used to make the function resilient to the same key piece being provided multiple times, and to account for multiple keys being from the same circle.

Key format

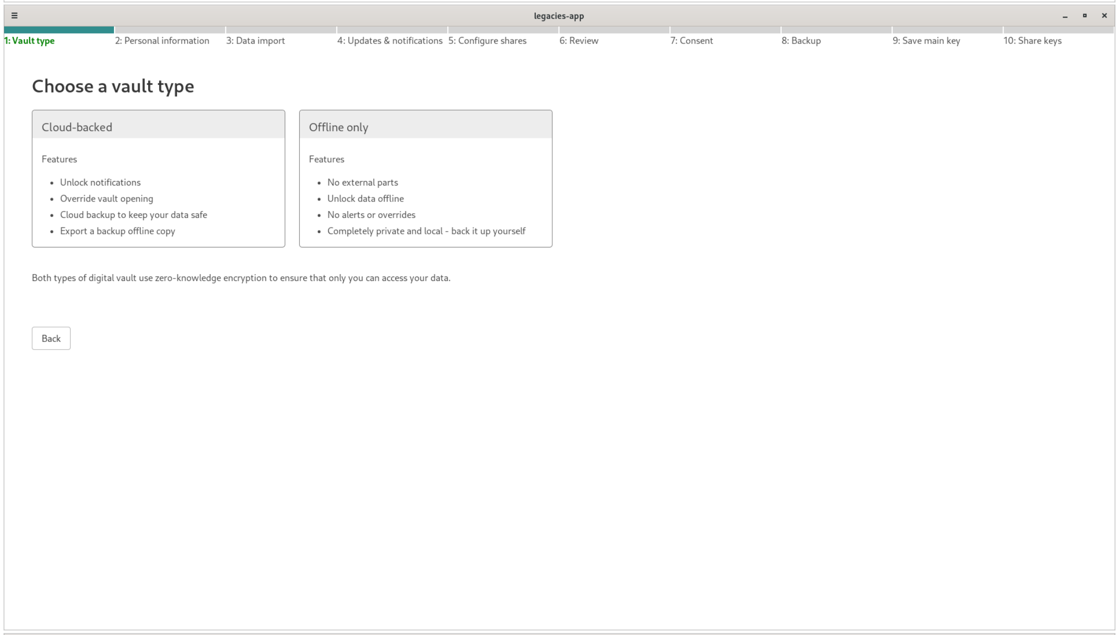
Once we have the keys generated by the above algorithm a key format is needed so that they can be shared and interpreted - Figure 14 shows the byte-level format of keys. A main key will only contain the owner cloud credential if it is a cloud backed vault (a cloud credential is a random 16 byte token generated in the desktop application, used to authenticate the data subject with the cloud provider), and a key piece will only include a circle key if it is a part of a required circle. The key distinction between a main key and a key piece is that a main key belongs to a data subject and has read-write access by itself, while key pieces must be combined to gain read-only access to the data. key pieces are a split of the main key produced by the key-splitting process defined in Figure 12.

All keys are hex-encoded and returned to the user in colon-seperated pairs. I chose this format to minimise the size of the keys, and so that the user has the option to export keys in plaintext (to clipboard) or as a file. Other key formats were considered, such as the mnemonic code approach of Bitcoin Improvement Proposal 39 [36] which gives a sequence of words, and is much easier to remember and use as a result. The BIP



Welcome

What would you like to do?



Choose a vault type

Cloud-backed

Features

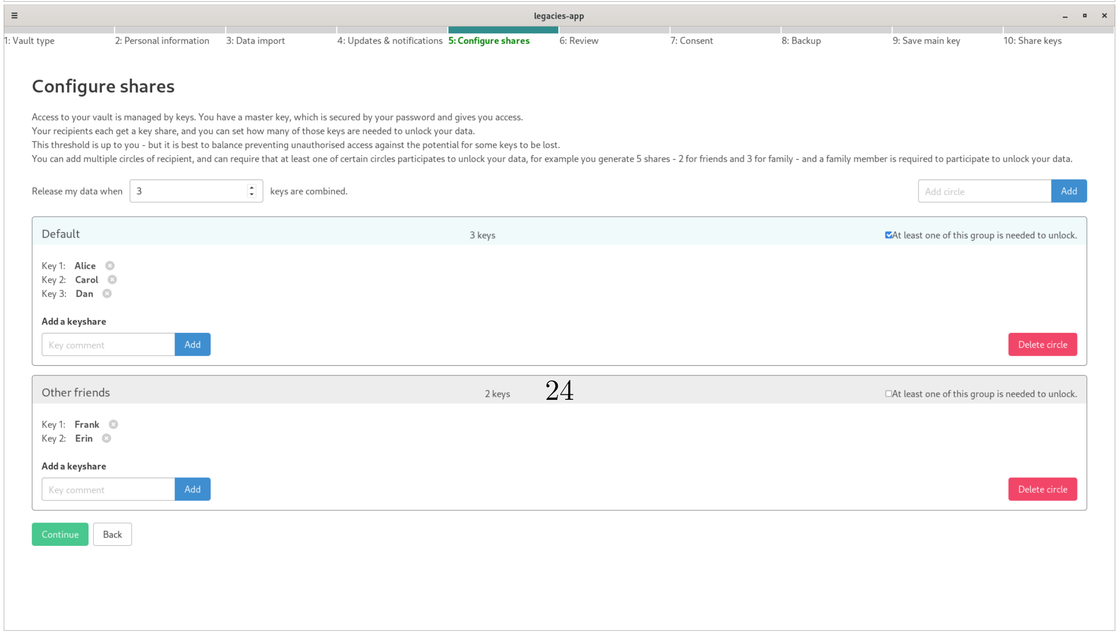
- Unlock notifications
- Override vault opening
- Cloud backup to keep your data safe
- Export a backup offline copy

Offline only

Features

- No external parts
- Unlock data offline
- No alerts or overrides
- Completely private and local - back it up yourself

Both types of digital vault use zero-knowledge encryption to ensure that only you can access your data.



Configure shares

Access to your vault is managed by keys. You have a master key, which is secured by your password and gives you access.
Your recipients each get a key share, and you can set how many of those keys are needed to unlock your data.
This threshold is up to you - but it is best to balance preventing unauthorised access against the potential for some keys to be lost.
You can add multiple circles of recipient, and can require that at least one of certain circles participates to unlock your data, for example you generate 5 shares - 2 for friends and 3 for family - and a family member is required to participate to unlock your data.

Release my data when 3 keys are combined.

Default 3 keys ☑ At least one of this group is needed to unlock.

Key 1: **Alice**
Key 2: **Carol**
Key 3: **Dan**

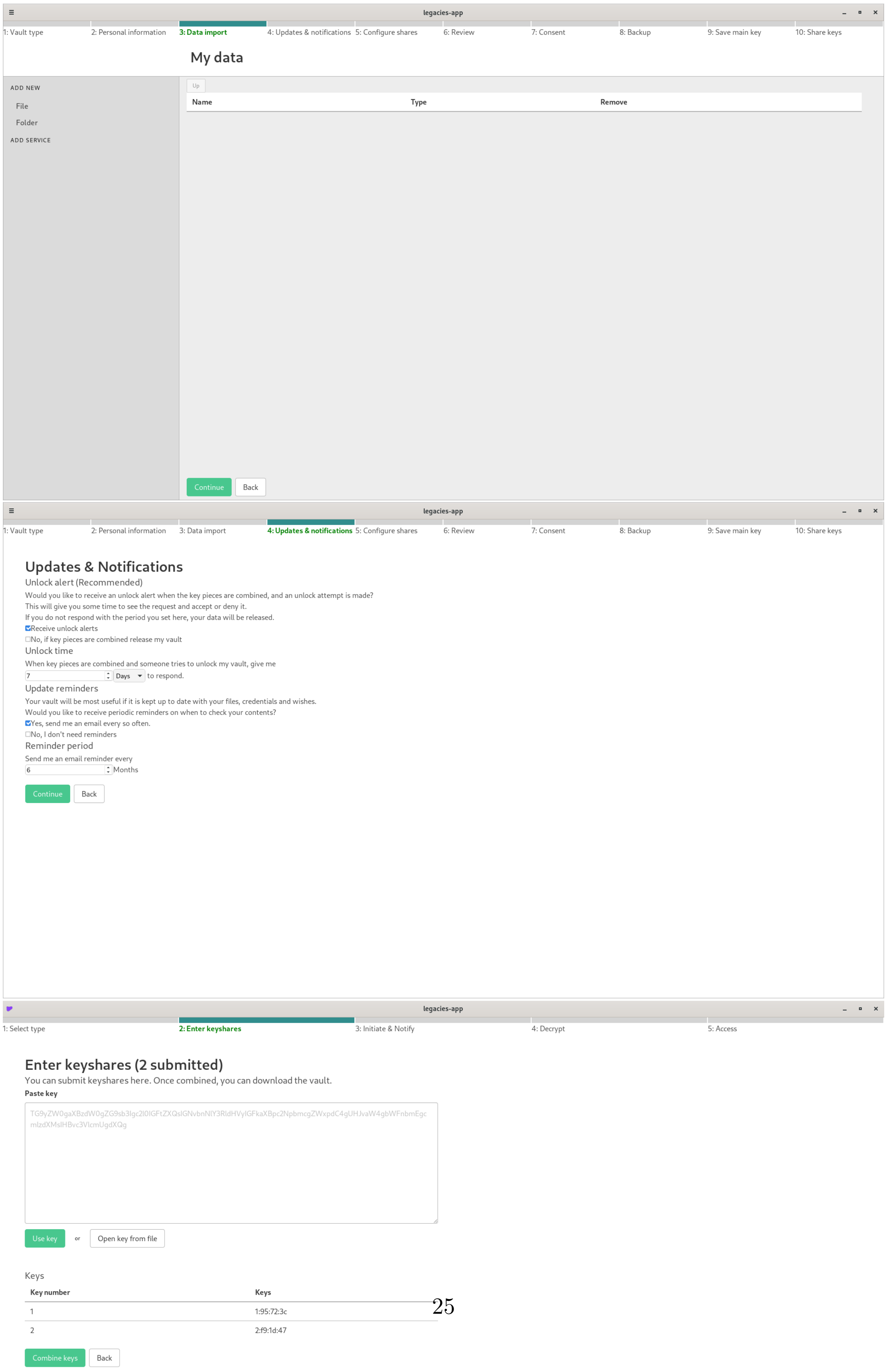
Add a keyshare

Other friends 2 keys ☐ At least one of this group is needed to unlock.

Key 1: **Frank**
Key 2: **Erin**

Add a keyshare

Figure 10: Interface screenshots



legacies-app

1: Vault type | 2: Personal information | **3: Data import** | 4: Updates & notifications | 5: Configure shares | 6: Review | 7: Consent | 8: Backup | 9: Save main key | 10: Share keys

My data

ADD NEW
File
Folder
ADD SERVICE

Up

Name	Type	Remove

Continue Back

legacies-app

1: Vault type | 2: Personal information | 3: Data import | **4: Updates & notifications** | 5: Configure shares | 6: Review | 7: Consent | 8: Backup | 9: Save main key | 10: Share keys

Updates & Notifications

Unlock alert (Recommended)

Would you like to receive an unlock alert when the key pieces are combined, and an unlock attempt is made?
This will give you some time to see the request and accept or deny it.
If you do not respond with the period you set here, your data will be released.

☑ Receive unlock alerts
☐ No, if key pieces are combined release my vault

Unlock time

When key pieces are combined and someone tries to unlock my vault, give me
7 Days to respond.

Update reminders

Your vault will be most useful if it is kept up to date with your files, credentials and wishes.
Would you like to receive periodic reminders on when to check your contents?

☑ Yes, send me an email every so often.
☐ No, I don't need reminders

Reminder period

Send me an email reminder every
6 Months

Continue Back

legacies-app

1: Select type | **2: Enter keyshares** | 3: Initiate & Notify | 4: Decrypt | 5: Access

Enter keyshares (2 submitted)

You can submit keyshares here. Once combined, you can download the vault.

Paste key

TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gUHJvaW4gbWFnbmEgc
mlzdXMsIHBvc3VlcmUgdXQg

Use key or Open key from file

Keys

Key number	Keys
1	1:95:72:3c
2	2:f9:1d:47

Combine keys Back

Figure 11: More interface screenshots

```
Given:
    M: Number of required keys to unlock.
    N: Total number of keys in the system.
    R: The number of required groups (<= N)
    S: Secret key - the actual key to unlock.

    Z: Individual share
```

1. Linear split S with M+1 of M=1 needed (linear/all keys needed)
2. Assign each L_n ($n \in 0..M-1$) to a circle. The final share L_M is Z, the individual share.
3. Threshold split Z using M of N into z_n ($n \in 0..N-1$)
4. Issue keys: Each member of a required circle gets their circle key L_M and one share of Z (z_n).
5. Each non-required gets a share of Z z_n only.

Figure 12: Key generation scheme for key-shares split into circles of trust.

wordlist could not be used as keys could have a length of up to 113 bytes, which with the BIP wordlist of 2048 words would require 83 words to represent the key - a significant increase in key length. When saved as a file, the main key file is given the extension .key and key pieces .ks. These extensions were chosen arbitrarily but provide a helpful distinction between purposes of each key to the user.

Figure 15 shows a high level overview of the vault file format. The meta information is encoded at the start as a MessagePack [37] and contains the following information:

- data subject name
- data subject email address
- circle data including whether they are required, circle name and the key comments.
- Nonce used for encryption

This is the Additional Authenticated Data (AAD) used by the AEAD algorithm, and it is signed to ensure integrity. I chose to include it as public meta information as it means the application can display this information (owner name, email and no keys required) when unlocking or opening an offline vault (see Figure 16) and as it makes it possible to identify the owner of a vault without knowing the keys.

```
Given:
    K: Array of full key pieces
```

1. Let I be a set of individual keys and C be a set of circle keys
2. For each key K_n of K, take the first 113 bytes as an individual key and add them to I
3. If the key K_n has a length greater than 113, take the next 33 bytes and add them to C
4. Combine the keypieces in I to get the individual key Z
5. Combine the keypieces in C with the individual key Z to get the decryption key k, and return that value.

Figure 13: Decryption algorithm given an array K of full keys

7.2 Cloud provider

The cloud provider is implemented using ActixWeb, SQLite and Rust and allows vaults to be uploaded, downloaded, updated and deleted - the owner's cloud key is used as a private key to uniquely identify a vault so a single owner can have many vaults.

While the design refers to a cloud provider that includes an email service and external cold-storage provider (Figure 1), due to time constraints, a reduced implementation has been used. From the user perspective there is no difference between the two approaches, and it still illustrates the same concept. In a system which uses a cold-storage provider, the cloud provider maintains basic state and loads the encrypted data from the provider as required, while in the implemented system the encrypted vault data is loaded directly into the database. I chose to use SQLite as it is very easy to use and works well for example purposes, and it would be well-suited to a system with a cold-storage provider. Moving to a proper database system would not be difficult due to the use of SQL to interact with the database. The cloud provider provides the basic required functionality, and I have designed it to be deployed behind a reverse proxy which would handle rate limiting, encryption and compression - if these features are required in a real-world environment.

The cloud provider maintains the following state about each vault it stores:

- data subject name and email address (used to send unlock alerts & identify their data)
- The encrypted vault data itself
- unlock_cloud_secret: An 8-byte secret key used to retrieve a copy of the encrypted

```
Main key
0                 32                48
+--------+--------+--------+--------+
|     Crypto      |   Cloud cred    |
|     Key (32)    |   (opt) (16)    |
+--------+--------+--------+--------+

Key piece
0                 113               146
+--------+--------+--------+--------+
| Individual Key  |   Circle key    |
| Z_n (113)       |   (opt) (33)    |
+--------+--------+--------+--------+
```

Figure 14: Key formats They encoded as Hex pairs when displayed to the user. This diagram shows the bytes along the top. (opt) indicates an optiona value.

```
Vault file format
0        4                          n                                          m
+--------+--------+--------+--------+--------+--------+--------+--------+-----...+
|Meta (4)| Meta_Data                |  Encrypted Vault contents                ...|
|Length  | Variable len(MsgPack)    |  Encrypted tar file (opaque binary data)...|
+--------+--------+--------+--------+--------+--------+--------+--------+-----...+
```

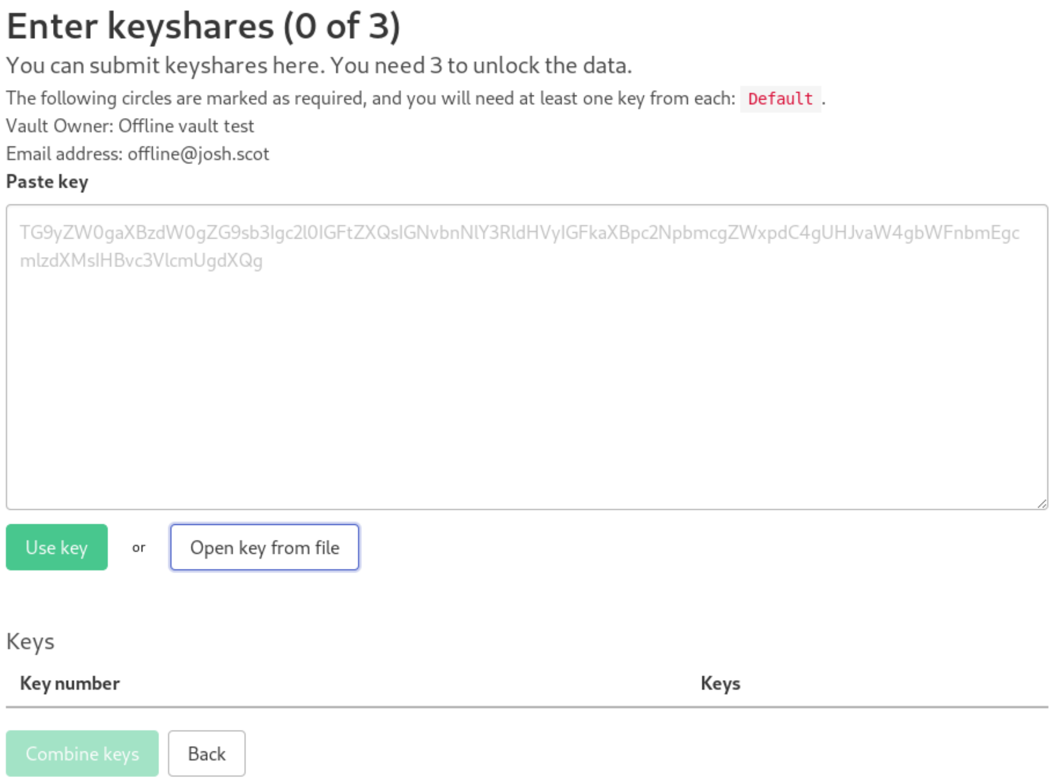
Figure 15: The vault file format

data. This is the first 8-byte of the decryption secret obtained by combining key-shares.

- owner_cloud_secret: A 16-byte secret key used to upload, update and delete the vault as required.
- reminder_period: The time between reminders, in days. A 0 value means it is disabled.
- alert_duration: The unlock notification grace period in seconds. A 0 value means it is disabled.
- open_at: If an unlock request has been made, at the unix timestamp at which to release the data.

One notable feature here is that the secret provided by a data recipient to retrieve the encrypted data file is a part of the key. It is important that the vault provider does not have access to the key - as it could then, theoretically, decrypt the data. Exposing the first 8 bytes is acceptable given that the cloud provider is semi-trusted, and current best practice suggests a key size of 24 bytes is sufficient, meaning the rest of the key remaining secret is sufficient to ensure security.

Ideally, this disclosure would not be necessary, but it is necessary due to the key-sharing process. The first key split described in Figure 12 produces key pieces of length



Enter keyshares (0 of 3)

You can submit keyshares here. You need 3 to unlock the data.

The following circles are marked as required, and you will need at least one key from each: `Default` .

Vault Owner: Offline vault test

Email address: offline@josh.scot

Paste key

TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gUHJvaW4gbWFnbmEgc
mlzdXMsIHBvc3VlcmUgdXQg

Keys

Key number	Keys

Figure 16: Screenshot showing the key piece input screen for an offline vault.

33 bytes, which can then be put into a secret sharing function which generates shares of length 113 bytes (with integrity and other properties). The keyshare function requires the input to be exactly 32 bytes so unlike the main key where a cloud secret is appended to the end, the first 8 bytes must be used. If necessary, for example due to advances computing power, the first 8 bytes can also be kept secret by adding an encrypted, separate cloud credential to all keys. The application could then access that credential by combining key pieces and decrypting that content.

7.2.1 Unlock process for cloud-backed vaults

Unlock attempts, involve a two-stage unlock process, shown in the diagram in Figure 17 where the application first makes a download request and then either waits for the alert notification to expire or proceeds to download the data itself.

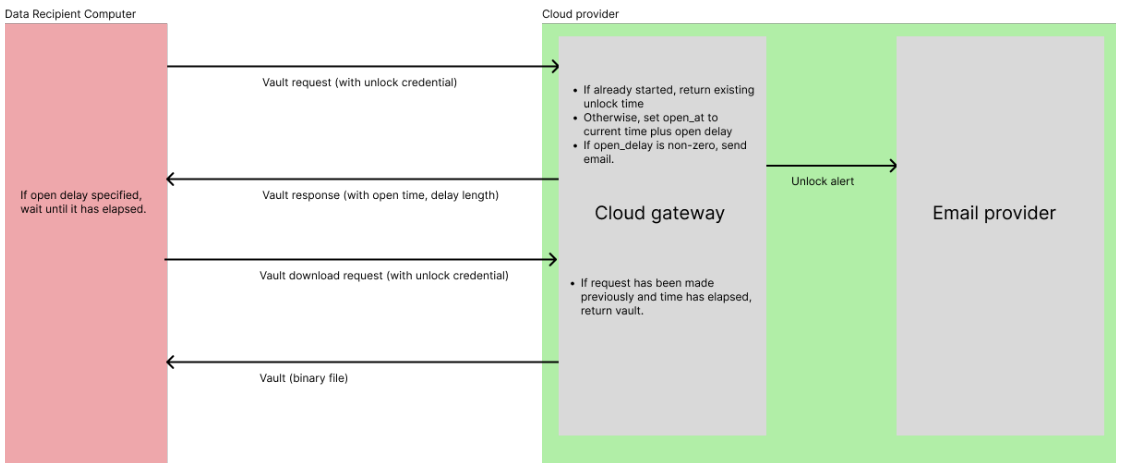
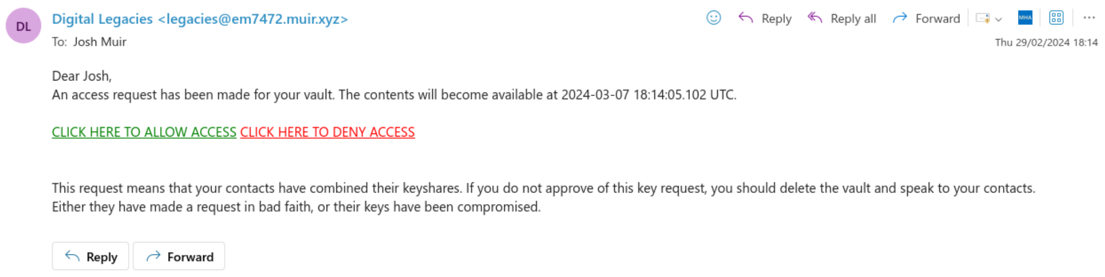

Data Recipient Computer
Cloud provider
Vault request (with unlock credential)
If already started, return existing unlock time
Otherwise, set open_at to current time plus open delay
If open_delay is non-zero, send email.
Unlock alert
If open delay specified, wait until it has elapsed.
Vault response (with open time, delay length)
Cloud gateway
Email provider
Vault download request (with unlock credential)
If request has been made previously and time has elapsed, return vault.
Vault (binary file)

Figure 17: Diagram showing the vault unlock flow including http requests and responses, and the basic logic in each participant.

When a request is made, as indicated in Figure 17, an email is sent with an unlock notification if the unlock period is enabled. Figure 18 shows the notification email, and gives the user links to approve or deny the request. If a request is denied, the user is expected to then delete their vault by opening the application or confer with their data recipients, as a new request could still be initiated using the keys. The alert functionality uses an external SMTP server, configured using environment variables. For the purposes of testing and developing this functionality, an existing SMTP server was used.

7.3 Testing

Testing was performed on an ongoing basis as features were implemented, and the desktop application also includes unit tests. Continuous integration was used on pull requests to identify regressions and prevent broken code being added to the main repository, and


DL Digital Legacies <legacies@em7472.muir.xyz>
To: Josh Muir
Thu 29/02/2024 18:14

Dear Josh,
An access request has been made for your vault. The contents will become available at 2024-03-07 18:14:05.102 UTC.

CLICK HERE TO ALLOW ACCESS CLICK HERE TO DENY ACCESS

This request means that your contacts have combined their keyshares. If you do not approve of this key request, you should delete the vault and speak to your contacts.
Either they have made a request in bad faith, or their keys have been compromised.

Figure 18: Screenshot showing the unlock notification email and links to further action

a systematic testing session (covering creation, open, unlock of all types using a test table) was performed, evidenced in Section A.1.

These unit tests cover the core functionality including cryptography and key sharing, the commands that provide a link between the user interface and rust code and meta parsing functions.

These tests were essential and helped identify some bugs with the implementation of the key-splitting algorithm, where the same key piece would be given out twice in certain configurations - resulting in failed decryption. Other testing included running the application on multiple machines and exchanging the generated vault files and keys to ensure that the decryption truly worked as expected - and not that the application was re-using the imported files/data under the hood.

Further testing was performed as part of the user study and some edge cases were found, such as a missing back button and issues where inconsistencies in file handling presented unreliable behaviour.

8 Evaluation and critical appraisal

8.1 User study

I undertook a user study to evaluate the application's usefulness and identify areas for improvement. A full description of the method for the user study is available in Appendix A.3. The method was designed to minimise learning effects and gauge user understanding across the different user scenarios - both as a person securing their digital legacy and as a data recipient trying to access a vault.

In the study, users were asked to carry out a number of tasks using the interface and then asked to answer a questionnaire covering their legacy planning and the application. 11 participants took part.

8.1.1 User study results

The headline results from the user study are that 9 out of 11 participants (82%) said they would be likely or very likely to use a tool like the project to manage their digital legacy, despite only one participant having done any previous legacy planning. Secondly, 8 of 11 participants (73%) rated the application as easy or very easy to use, while the rest rated it as neutral, indicating that the application is user-friendly. Notably, most participants (6) were not aware of the concept of digital legacy at all, and of those that were aware (5), none had made any plans. A possible cause is that the participants are dealing with their digital legacy by neglecting it - a known coping mechanism [3].

Looking in more depth at user opinions for different scenarios, Figure 19 shows a comparison of user-reported difficulty for the application based on the first scenario/condition to which participants were exposed. It shows that users reported a higher level of ease using the application where they had an offline condition first, which suggests that the offline configuration was easier to use. This may be because the offline opening process provides extra information to the user such as the name of the vault owner, number of key pieces needed and required circles.

Figure 20 shows the user-reported difficulty in cases where participants did a vault creation scenario first and where they did an unlock scenario first. This is counter to my expected result and shows users reported higher ease of use when they did the creation scenario second. A possible reason is that the creation scenario was easier to complete than unlocking, or they had built some familiarity with the interface before carrying out that task. This is an area of the application that merits further examination.

On the whole, the user study suggests the application is easy to use with the worst feedback received being 'neutral' for difficulty.

Participants were also asked for any improvements they would make, and if there were any concepts that were not clear. The concepts most rated as unclear were:

Cancellation period The term cancellation period, referring to the unlock alert and override was not understood by three participants. This is despite two of those participants having a condition that involved creating a cloud-backed vault.

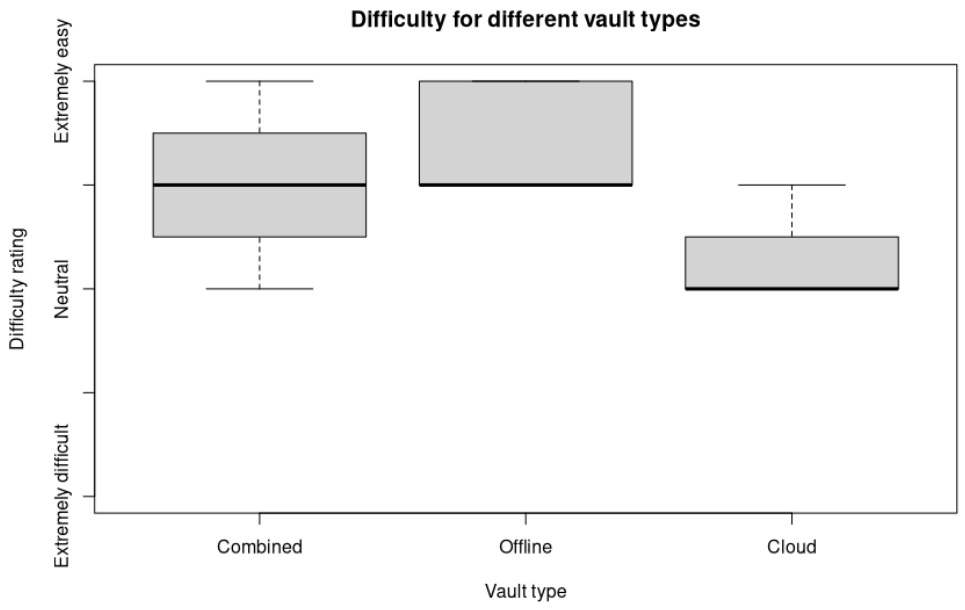


Figure 19: Boxplot showing a comparison of user-reported difficulty for all, users who did a cloud-backed scenario first, and users who did an offline scenario first.

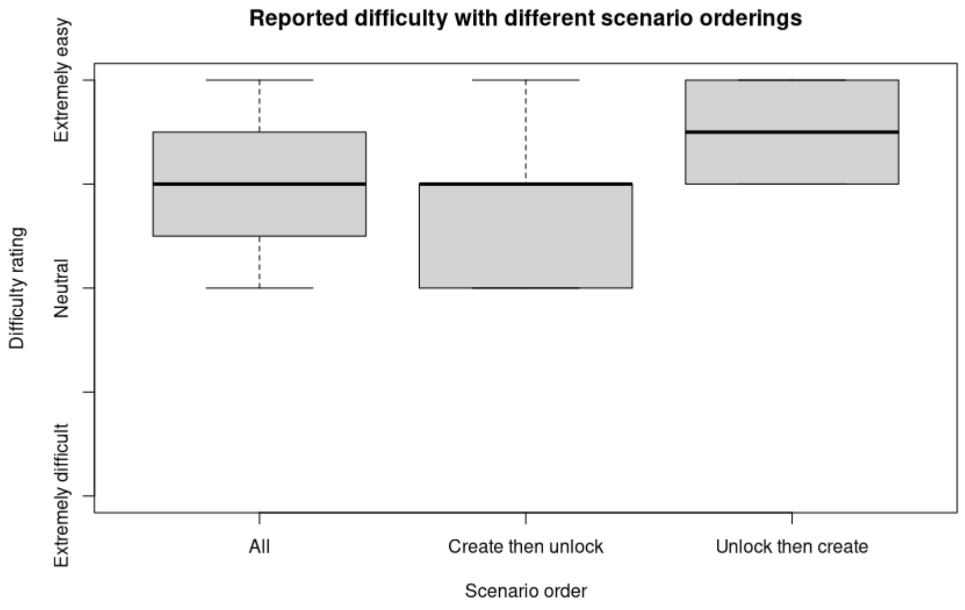


Figure 20: Boxplot showing a comparison of user-reported difficulty for all, users who did a creation scenario first and users who did an unlock scenario first.

Circle This concept was poorly understood, and suggests that the user interface may need to be updated to better explain the concept of circle and how it relates to social groups.

The above results suggest that a wider investigation into the 'circle' term from [34] may be appropriate to determine if it is the best term to use, or if another term such as 'group' is more suitable.

Limitations of user study

One limitation of the user study is that it focused on the core vault functionality, and while it tested the use of the cloud-backed vaults, unlock alerts were not tested, as they could not be feasibly tested in a single user study session.

Both Figures 19 and 20 have some unusual artifacts due to the low number of participants involved in the study. The boxplot showing the full difficulty range (left boxplot in both figures) is the only one that could be considered a typical boxplot, and this is because in Figure 19 in the offline boxplot (center), all participants rated the difficulty as either 'Somewhat Easy' or 'Extremely easy' with most reporting the difficult as somewhat easy. As a result, the middle boxplot does not feature whiskers, there is no lower quartile and the median is the bottom line of the boxplot. Similar issues affect the third boxplot in Figure 19 but it also has an outlier shown as it is within 1.5 times the very low interquartile range.

Another possible limitation of the user study lies in the participant-reported difficulty scores. Most of the recruited participants were friends or associates and as a result may have been tended towards giving good scores. I tried to control for this as can be seen in the user study method (A.3) by using an anonymous form instead of interviewing to gather difficulty data and leaving the participants on their own to complete the form. During the study 9 of 11 participants were able to complete all tasks successfully so the impact of this limitation may be limited - and the difficulty score an accurate reflection of the application.

8.1.2 Further work and changes made

Following the user study, the following changes were made to the application to improve ease-of-use:

- Some participants did not save their main key, so it is no longer possible to progress beyond the main key page without saving it.
- Erroneous references to master key were removed and replaced with main key.
- Default keyfile names were improved - now default to the 'key comment.ks'.
- Colours were updated to be more clear on low contrast screens.
- Text on the share configuration page was updated to more explicitly explain what a circle is.

- Enhancement to the unlock process means that the user can now choose the save location for the data output.
- Added explicit indication to the fields on the personal information page to indicate what is required and what is not.
- Sharing/key distribution was improved - Added send via. Email & send via. WhatsApp buttons.
- Fix bugs - Missing back button and creating a vault, opening it and then unlocking it previously failed for cloud vaults.

I made these fixes in cases where more than 1 participant identified the same issue. 9 out of 11 participants had issues identifying whether the detailed personal information fields were optional, so that fix was particularly important.

I chose to add WhatsApp key sharing support as WhatsApp is very widely used and end-to-end encrypted, so it can be considered a good compromise between security and usability for key distribution. Email sharing support also makes it easier to share keys - and it is likely to be the way users share their keys - but depending on their email provider and email habits it may be less secure. Despite this, I felt it was worth the increased usability for key distribution and the choice is ultimately up to the user.

There were some other improvements suggested by participants, including:

- Drag + Drop support for keys, and file association to allow a key to be double-clicked and opened with the application.
- Make the file prompts open in the last saved location rather than a default location.
- Import page buttons were hard to see as they blended in with sidebars - make them easier to see.
- Support circles and keys to be directly renamed by double-clicking on them, instead of needing to remove and re-add them.

These fixes could be undertaken as future work.

8.2 Objectives

In this section I will go through the primary, secondary and territory objectives set out in the DOER document and evaluate how well they were met.

First, as a summary:

- All primary objectives have been met.
- The first two secondary objectives have been met.
- The final secondary objective has not been met.
- Neither tertiary objective has been met but possible approaches to long-life media are discussed in this section.

Primary

The first primary objective - Allow files to be deposited into an encrypted digital vault has been met. The program allows users to deposit files and folders of their choosing into the digital vault, and delete them as required. During the user study, 9 of 11 participants were able to successfully deposit files and create an encrypted vault.

The second primary objective - Provide a sharing mechanism - has been met. The program generates a key and splits it into key pieces, which can then be shared as plaintext - through the clipboard - or in file format.

The third primary objective - Different levels of trust - has been met and is implemented through the circle concept. Circles can be set as required or not required, which determines whether at least one of the key-holders in that circle must participate in the final process. As a result, keys in a non-required circle are less trusted than those in a required circle.

The final primary objective - Make the system compliant with relevant privacy law - has been achieved through the creation of a Data Protection Impact Assessment and the inclusion of the consent page in the desktop application. The program supports the basic create, delete and update operations, and users are responsible for keeping their data up to date. The application is designed to comply with GDPR to ensure that user data is properly safeguarded both during life and following death.

Secondary

The first secondary objective - Use distributed systems or other technology to improve the resilience of the system - has been met but there is room for further expansion. The original design of the system included the use of AWS Glacier or a similar long-term cloud cold-storage solution to ensure the availability of vault data. This functionality has not been implemented, but vault files are instead backed up directly on the cloud backend instance, which provides similar assurances. Implementing this functionality would be a backend change with no visible change in behaviour to the user or desktop application.

The second secondary objective - to support cancellation functionality - has been achieved. The cloud provider sends an email when a new vault is created, and an email when a download request is made. This email contains links allowing the data subject to accept or decline the download request according to their wishes, or the data will automatically become available when the time-out expires. This behaviour is very similar to how other technologies implement emergency access or digital legacy features.

The final secondary objective was to implement a system to keep the vault up to date over time. This objective has not been achieved due to time constraints, but the basic tools required to implement it are in the system. It could be achieved by using a cron script which runs at the same time every day and checks for vaults with the reminder period set. If a reminder period is elapsed, an email could be sent to those data subjects reminding them to update their data and provide instructions on how to do so.

Tertiary

Neither Tertiary objective has been met. Examining the problem of ensuring digital vaults surviving the lifespan of a human, we can assume that the vault itself will outlast a lifespan if held in a cloud-backed format due to the use of cold-storage media. For the keys, possible solutions include printing them into dog tags [38] or printing them on laminated archive paper [39, Paper Choices and Storage].

8.3 Similar work

The most commonly used similar software is included in password managers. LastPass, BitWarden, Keeper, NordPass and RoboForm [22–25, 40] support a delay-based access request, where once a person has agreed to be a data recipient, they can request access to the data and receive it if no intervention is made within a set period - which in most cases is 7 days. The key limitation of these is that all require the recipients to create an account on the service, and be able to access to that account when the emergency access is required, presenting a significant risk that the data will be lost. A second drawback is that that emergency access is entirely dependent on the password manager companies continuing to operate. If a company becomes insolvent or is bought by a competitor, data may be lost. Using an offline vault does not suffer from this limitation.

There are other solutions which do not rely on an external company's servers and service continuing to operate such as Paperback [39] and SSKR [41].

Paperback uses a similar approach to this project, where Shamir's secret sharing is used to share a secret but both the data subject's personal data and all key pieces are encoded in QR-code form so that they can be printed. SSKR is less relevant and is focused on using secret sharing and an easy way of encoding keys to securely share and preserve the key to a bitcoin wallet. However, the same software and principles could be used to share any key. One less widely used tool is Hereditas [42] which seeks to solve the same problem of digital legacy using digital vaults, but does use not secret sharing and relies on the existence of a web server. That may be a poor assumption to make given following the death of the data subject, any ongoing billing or infrastructure is likely to fail.

Existing solutions can be considered on a spectrum from entirely service-based to entirely local, and this project exists in the middle - users have the option of introducing an external dependency (the cloud provider) to get the additional features and data safety it provides, or they can use a fully local solution. This enhances user control of their own data. Paperback, while being the most comparable existing software, is primarily suited to technical users as the interface is command-line based, unlike this project which seeks to support average users to curate and secure their digital legacy.

9 Conclusions

I have created a new mechanism to secure, share, and manage digital legacy in a way that maximises privacy and user control while remaining easy to use. It broadens the usability of secret-sharing algorithms and uses trusted cryptographic algorithms, allowing the public to benefit from technology previously accessible only to technical users.

It allows users to curate their digital legacy and to secure and share the data and credentials they want to pass on. The key achievements of this project are the creation of the desktop application to secure and share digital legacies, the cloud provider to provide data safety assurances and unlock alerts, the circles concept to provide multiple levels of key trust, and the DPIA to assess the privacy impact of the system. I also gathered the opinions of some end users towards digital legacy and the application itself - which suggested that most users found it easy to use.

Some of the key drawbacks of my work include that a cold-storage service was not integrated into the cloud provider, and key distribution remains difficult despite improvements made following the user study. Adding the option to share through email and WhatsApp makes distribution easier but still places an onus on the user to understand the principles of secure key distribution, something that could be improved with further work. Another drawback is that the desktop application relies heavily on buffers and temporary files, and this introduces both a bound on the maximum size of the archive (as the entire archive is held in memory at once) and means over time, temporary files will build up on the file system.

9.1 Further work

There is much room for future work and avenues that could be explored. Future work for the system itself could include adding support for update reminders, interoperating with online platforms, improving the key format, using a cold cloud storage provider and using stream-cipher based cryptography to support larger vault archives. The key format improvements could involve adding the key comment to the key file, instructions and a signature to ensure file integrity and A similar key format to SSH keys [43] could be used.

The application could also be updated to use stream-based cryptography so that the vault can be encrypted incrementally - and does not need to be held entirely in memory. Security for the main key could be improved by adding a password to the main key or using the trusted platform module (TPM) on the system to store it, if available.

Interoperability is a particularly difficult problem to solve, as in most cases, it cannot be done using existing APIs without violating the platform's terms of service. Platforms such as Facebook or Google do not permit access to accounts once the account holder passes away, as the terms of service is an agreement between the platform and the holder only while they are living.

This would mean that any action, for example, automated deletion when the vault is opened, would fall foul of those policies.

As mentioned previously, key distribution is another area for further work and could be improved using a print facility. This could be a system to print the keys in an easy-to-use and long-lasting format, such as the QR code sheets used by paperback [39] or by the word-based backup method employed by SSKR [41], with SLIP-0039 [44] being another similar alternative.

References

[1] E. Harbinja, *Legal Aspects of Transmission of Digital Assets on Death.* PhD thesis, University of Strathclyde, 2017.

[2] "A look at Password Health Scores around the world in 2022." https://ripleyprd.wpengine.com/global-password-health-score-report-2022.

[3] J. Pfister, ""This will cause a lot of work.": Coping with Transferring Files and Passwords as Part of a Personal Digital Legacy," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, (New York, NY, USA), pp. 1123–1138, Association for Computing Machinery, Feb. 2017.

[4] DeviantOllam, "Lawyer. Passport. Locksmith. Gun. (A Talk About Risk & Preparedness)." https://www.youtube.com/watch?v=6ihrGNGesfI, Nov. 2022.

[5] C. J. Öhman and D. Watson, "Are the dead taking over Facebook? A Big Data approach to the future of death online," *Big Data & Society*, vol. 6, p. 205395171984254, Jan. 2019.

[6] Google LLC, "About Inactive Account Manager - Google Account Help." https://support.google.com/accounts/answer/3036546?hl=en.

[7] Apple Inc., "How to add a Legacy Contact for your Apple ID." https://support.apple.com/en-gb/HT212360, Sept. 2023.

[8] Apple Inc., "How to request access to a deceased family member's Apple account." https://support.apple.com/en-gb/HT208510, Sept. 2023.

[9] Meta Platforms, Inc., "About memorialised accounts | Facebook Help Centre." https://en-gb.facebook.com/help/1017717331640041.

[10] E. Harbinja, L. Edwards, and M. McVey, "Governing ghostbots," *Computer Law & Security Review*, vol. 48, p. 105791, 2023.

[11] "Data Protection Act 2018," 2018.

[12] H. Antoine, "Digital Legacies: Who Owns Your Online Life After Death?.," *Computer & Internet Lawyer*, vol. 33, no. 4, pp. 15–20, 2016.

[13] EUROPEAN PARLIAMENT AND OF THE COUNCIL, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)," May 2016.

[14] E. Harbinja, "Emails and death: Legal issues surrounding post-mortem transmission of emails," *Death Studies*, vol. 43, pp. 435–445, Aug. 2019.

[15] E. Harbinja, “Post-mortem privacy 2.0: Theory, law, and technology,” *International Review of Law, Computers & Technology*, vol. 31, pp. 26–42, Jan. 2017.

[16] U. Kohl, “What post-mortem privacy may teach us about privacy,” *Computer Law & Security Review*, vol. 47, p. 105737, 2022.

[17] R. N. Nwabueze and H. Hancock, “What’s wrong with death images? Privacy protection of photographic images of the dead,” *Computer Law & Security Review*, vol. 47, p. 105715, 2022.

[18] C. Fairbairn, “Obtaining a copy of a will,” Research Briefing 03194, House of Commons Library, Oct. 2022.

[19] R. Anderson, 1956, “Security engineering : A guide to building dependable distributed systems / Ross J. Anderson.,” pp. 0–1, Tue Oct 03 15:58:22 BST 2023.

[20] A. Shamir, “How to Share a Secret,” *Commun. ACM*, vol. 22, pp. 612–613, Nov. 1979.

[21] “Build smaller, faster, and more secure desktop applications with a web frontend | Tauri Apps.” https://tauri.app/.

[22] “Emergency Access | Bitwarden Help Center.” https://bitwarden.com/help/emergency-access/.

[23] “Emergency Access - LastPass.” https://www.lastpass.com/features/emergency-access.

[24] “Emergency Access.” https://docs.keeper.io/user-guides/emergency-access.

[25] R. Rawlings, “New NordPass Feature Is Here — Emergency Access.” https://nordpass.com/blog/introducing-emergency-access/, Nov. 2021.

[26] “What is a DPIA? | ICO.” https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/accountability-and-governance/data-protection-impact-assessments-dpias/what-is-a-dpia/#what1.

[27] “Figma: The Collaborative Interface Design Tool.” https://www.figma.com/.

[28] J. Nielsen, “10 Usability Heuristics for User Interface Design.” https://www.nngroup.com/articles/ten-usability-heuristics/, Apr. 1994.

[29] “Structuring forms - Service Manual - GOV.UK.” https://www.gov.uk/service-manual/design/form-structure.

[30] “Button.” https://design-system.service.gov.uk/components/button/.

[31] V. Friedman, “Designing A Better Back Button UX.” https://www.smashingmagazine.com/2022/08/back-button-ux-design/, Aug. 2022.

[32] "Bulma: Free, open source, and modern CSS framework based on Flexbox." https://bulma.io.

[33] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," Request for Comments RFC 7539, Internet Engineering Task Force, May 2015.

[34] C. Allen, "Shamir Secret Sharing Best Practices," Mar. 2019.

[35] B. Schneier, "Crypto-gram: October 15, 1998 - Schneier on Security." https://www.schneier.com/crypto-gram/archives/1998/1015.html#cipherdesign, Oct. 1998.

[36] M. Palatinus, P. Rusnak, A. Voisine, and S. Bowe, "Mnemonic code for generating deterministic keys." https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki.

[37] "MessagePack: It's like JSON. but fast and small.." https://msgpack.org/index.html.

[38] "Crypto-commons/Docs/sskr-cold-storage.md at master · BlockchainCommons/crypto-commons." https://github.com/BlockchainCommons/crypto-commons/blob/master/Docs/sskr-cold-storage.md.

[39] A. Sarai, "Cyphar/paperback," Mar. 2024.

[40] "What is Emergency Access and how does it work?." https://help.roboform.com/hc/en-us/articles/115005664827-What-is-Emergency-Access-and-how-does-it-work.

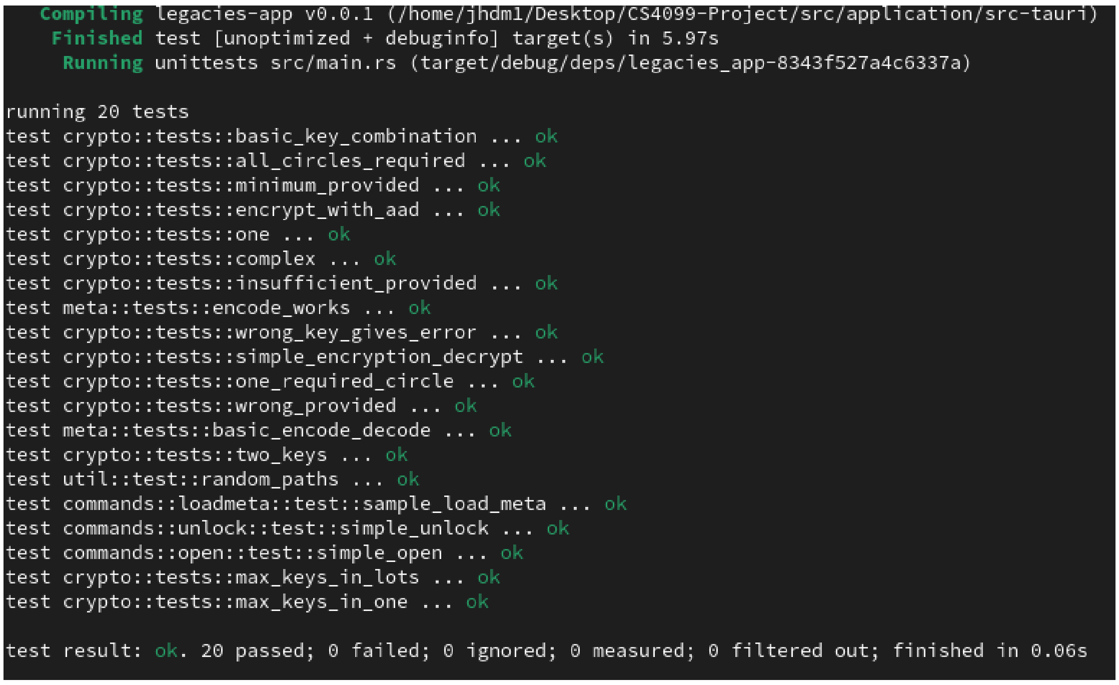
[41] "Crypto-commons/Docs/sskr-users.md at master · BlockchainCommons/crypto-commons." https://github.com/BlockchainCommons/crypto-commons/blob/master/Docs/sskr-users.md.

[42] A. A. Segala, "ItalyPaleAle/hereditas," Feb. 2019.

[43] R. L. Thayer and J. Galbraith, "The Secure Shell (SSH) Public Key File Format," Request for Comments RFC 4716, Internet Engineering Task Force, Nov. 2006.

[44] P. Rusnak, A. Kozlik, O. Vejpustek, T. Susanka, M. Palatinus, and J. Hoenicke, "Shamir's Secret-Sharing for Mnemonic Codes," Dec. 2017.

[45] "What is zero-knowledge cloud storage?." https://proton.me/blog/zero-knowledge-cloud-storage, June 2023.



```
   Compiling legacies-app v0.0.1 (/home/jhdm1/Desktop/CS4099-Project/src/application/src-tauri)
    Finished test [unoptimized + debuginfo] target(s) in 5.97s
     Running unittests src/main.rs (target/debug/deps/legacies_app-8343f527a4c6337a)

running 20 tests
test crypto::tests::basic_key_combination ... ok
test crypto::tests::all_circles_required ... ok
test crypto::tests::minimum_provided ... ok
test crypto::tests::encrypt_with_aad ... ok
test crypto::tests::one ... ok
test crypto::tests::complex ... ok
test crypto::tests::insufficient_provided ... ok
test meta::tests::encode_works ... ok
test crypto::tests::wrong_key_gives_error ... ok
test crypto::tests::simple_encryption_decrypt ... ok
test crypto::tests::one_required_circle ... ok
test crypto::tests::wrong_provided ... ok
test meta::tests::basic_encode_decode ... ok
test crypto::tests::two_keys ... ok
test util::test::random_paths ... ok
test commands::loadmeta::test::sample_load_meta ... ok
test commands::unlock::test::simple_unlock ... ok
test commands::open::test::simple_open ... ok
test crypto::tests::max_keys_in_lots ... ok
test crypto::tests::max_keys_in_one ... ok

test result: ok. 20 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.06s
```

Figure 21: Screenshot showing the result of running all 20 Rust unit tests

A Appendices

A.1 Testing

A.1.1 Unit tests

The core functionality of the Desktop application has unit tests in place. These tests cover encryption, secret sharing and re-combination, and the core commands used by the user interface to verify that they function as expected. I did not add unit tests for the frontend as I felt it could be more efficiently tested manually, as described in the tables below. I also did not write tests for the cloud provider as it is very simple and the majority of the code is oriented around updating the database. It is included in the manual tests in section A.1.2.

The unit tests for the Rust functionality were very useful to find edge cases with the encryption, such as an edge case with the key splitting algorithm that resulted in the same individual keyshare being issued to multiple key pieces - meaning the keys could not be recombined and the data could not be decrypted. The tests also include edge cases such as requiring 1 of 1 keyshare or 1 of 2 and similarly small numbers - as well as the maximum - 256.

In accordance with Rust conventions, unit tests are included in a test module in the same file as the code they are testing. Figure 21 shows the output of a successful test run.

A.1.2 Manual testing

The manual testing focused on the user interface - both on realistic values and on extreme values to see how the application responded. The user study also presented an opportunity for manual testing under real-use conditions and participants found some issues (discussed in the evaluation) such as missing back buttons. This testing was performed after the completion of the user study, Table 2 shows the main tests performed.

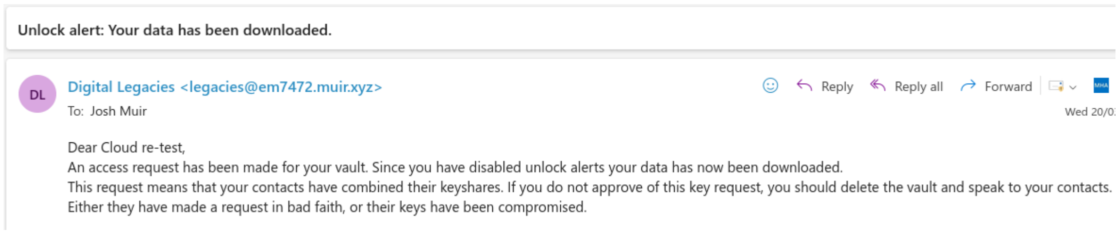
Table 2: Test table for overall system tests.

Test	**Expected output**	**Actual Output**	**Comments**
Create offline vault	Application creates vault App does not allow progression until vault file saved.	Vault created.	See Figure 24 for the key copy interface.
Create cloud-backed vault	Application creates vault and uploads to cloud. Email notification sent. Offline download option available.	Vault created.	
Open offline vault (Main key)	App asks for file and main key and opens as expected. User can choose where to save.	Vault opens.	
cloud-backed vault (Main key)	App asks for main key and opens as expected.	Vault is downloaded and opens.	
Unlock offline vault (Using key pieces)	App asks for vault file and then key pieces. UI Displays the number required.	Key pieces screen shows number needed. Vault unlocks as expected.	See Figure 11 for key input page.

Continued on next page

Table 2: Test table for overall system tests. (Continued)

Unlock cloud backed vault No alert period (Use key pieces)	App asks for key pieces and attempts to download and unlock data.	App downloads vault as expected. Alert email was sent to vault creator allowing them to approve/deny.	This email should not contain approve/deny buttons. A vault with no alert period cannot be have open requests Denied.
Cloud backed vault No alert period (Test 2) (use key pieces)	App asks for key pieces and attempts to download and unlock data. Email sent indicating data downloaded.	App asks for key pieces and attempts to download and unlock data. Email sent indicating data downloaded.	As expected - email is now accurate to what has happened. See Figure 22.
Unlock cloud vault on different machine	As above	As expected.	This shows the program works as expected, and does not apply on side-effects or left over files from the creation period. Unlocking and creation are entirely independent. See Figure 23
Unlock cloud vault with unlock notification enabled.	Unlock request made and UI shows time remaining. Data decrypted once time elapsed.	As expected. Data decrypted once time elapsed.	
Delete cloud-backed vault (Open with main key then click Delete)	Vault is deleted and user is returned to main screen.	Vault is deleted and user returned to main screen.	



Unlock alert: Your data has been downloaded.

Digital Legacies <legacies@em7472.muir.xyz>

To: Josh Muir

Wed 20/0

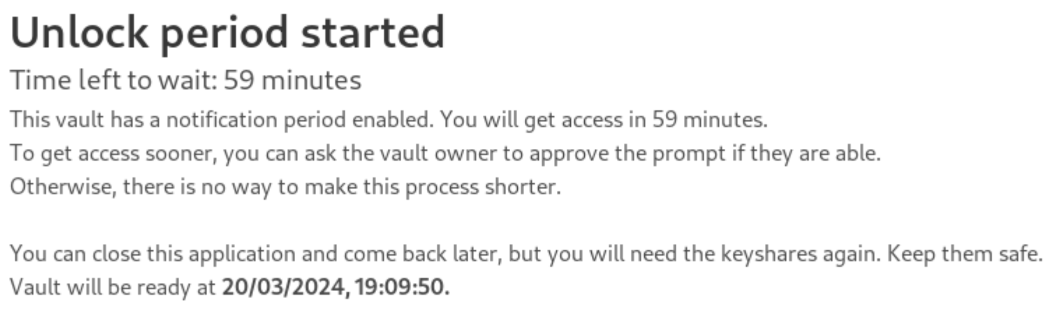
Dear Cloud re-test,

An access request has been made for your vault. Since you have disabled unlock alerts your data has now been downloaded.

This request means that your contacts have combined their keyshares. If you do not approve of this key request, you should delete the vault and speak to your contacts.

Either they have made a request in bad faith, or their keys have been compromised.

Figure 22: Screenshot showing the updated alert email for vaults with no alert notification enabled.



Unlock period started

Time left to wait: 59 minutes

This vault has a notification period enabled. You will get access in 59 minutes.

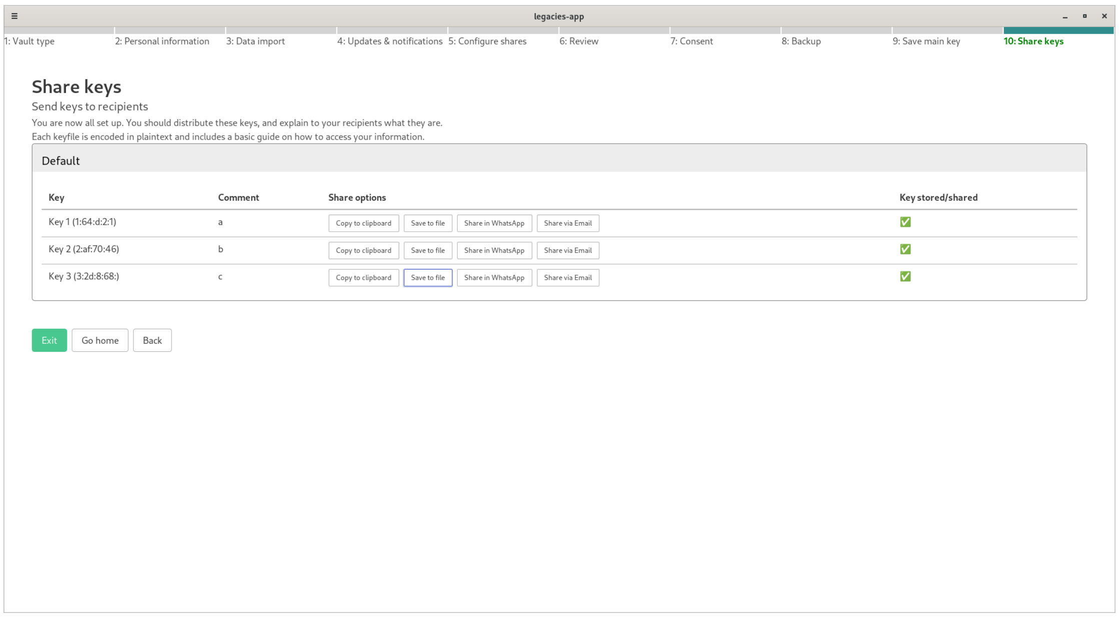
To get access sooner, you can ask the vault owner to approve the prompt if they are able.

Otherwise, there is no way to make this process shorter.

You can close this application and come back later, but you will need the keyshares again. Keep them safe.

Vault will be ready at **20/03/2024, 19:09:50.**

Figure 23: Screenshot showing the waiting page for a vault with an unlock notification enabled.



Share keys

Send keys to recipients

You are now all set up. You should distribute these keys, and explain to your recipients what they are.

Each keyfile is encoded in plaintext and includes a basic guide on how to access your information.

Default

Key	Comment	Share options	Key stored/shared
Key 1 (1:64:d:2:1)	a	Copy to clipboard / Save to file / Share in WhatsApp / Share via Email	✅
Key 2 (2:af:70:46)	b	Copy to clipboard / Save to file / Share in WhatsApp / Share via Email	✅
Key 3 (3:2d:8:68:)	c	Copy to clipboard / Save to file / Share in WhatsApp / Share via Email	✅

Figure 24: Screenshot showing the updated alert email for vaults with no alert notification enabled.

A.2 Manual

There are two software components: The Desktop application (in src/application) and the Cloud provider (in src/cloud-provider). In addition to this manual both of these folders contain README files (README.md) which explain how to build the components and provide an overview of key files.

A.2.1 Desktop application

The Desktop application can be built on the lab machines by navigating to src/application and running the following commands:

```
npm install
npm run tauri build
```

These commands will install the required dependencies - the system has some other dependencies but these are installed on all lab machines. Those dependencies (for Arch Linux) are:

- Rust 1.73.0 or later.
- webkit2gtk4.0-devel
- openssl-devel
- curl
- wget
- file
- libappindicator-gtk3-devel
- librsvg2-devel
- C Development Tools and Libraries

The only configuration required is that the BACKEND_URL in `src/application/src/app.tsx` may need to be changed.

Tauri means the desktop application can be built on all platforms, but the target must be specified. By default, it will build for the system it is currently on. For testing and demonstration purposes, the AppImage output was used. This format bundles all the required dependencies and is portable across most Linux systems.

A.2.2 Cloud provider

The Cloud-provider can be built on the lab machines by navigating to src/cloud-provider and running the following commands:

```
cargo build --release
```

Then the binary will be in target/release.

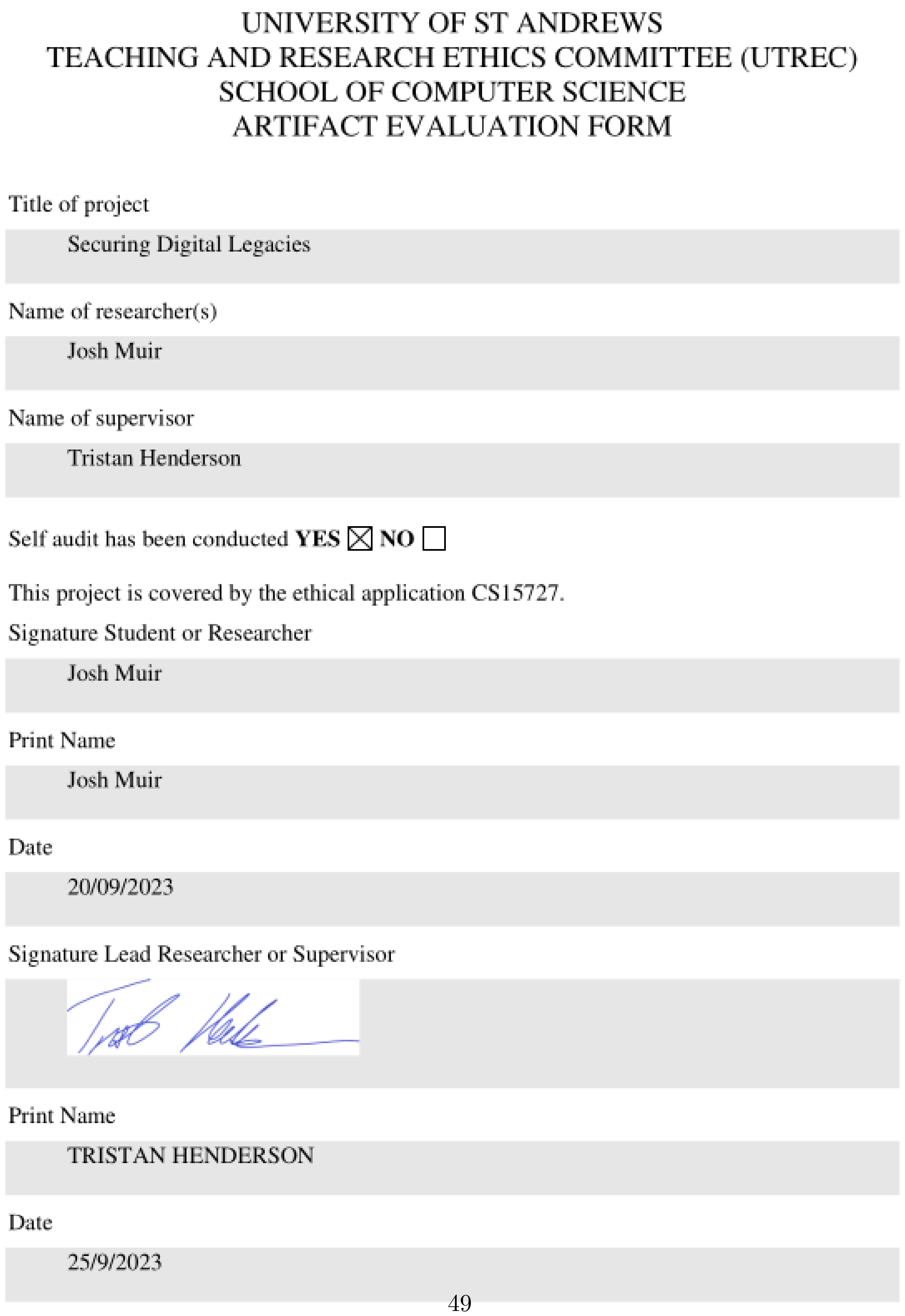
Unfortunately it is not currently possible to build the backend with the version of Cargo installed on the lab machines, as Version 1.73 or later is required. This can be solved by installing a later version of cargo manually using rustup, and setting the environment variable RUSTUP_HOME and adding the cargo binary to path. Cargo 1.76.0 was used during development and to build the releases of this software using this method.

The cloud provider will listen on port 22469 - my user id. nginx or a similar proxy can then be used to proxy requests to that port. Table 3 shows the environment variables that must be set. They cover configuration to send emails and alerts and environment is used as is best practice for secrets, and means the cloud provider can be easily configured without recompiling the binary.

Table 3: Environment variables for the cloud provider

Variable name	**Type**	**Description**
SMTP_USERNAME	String	Username to use to connect to SMTP server.
SMTP_PASSWORD	String	Password to use to connect to SMTP server.
SMTP_HOST	String (Hostname)	Hostname to connect to for SMTP. Uses port 587.
MAIL_FROM	String (Email)	Email address to put in the 'from' field. This can include the name to send from, i.e: "Digital Legacies" legacies@example.com.
BACKEND_URL	String (Hostname)	This is the public URL of the server. It should be the hostname only and the path to accept/deny requests will be appended onto the end. Only used in emails - has no bearing on what the server will listen on.
PORT	u16 (Optional)	Port number to listen on. Optional: If it is not provided (or is invalid), it will listen on 22469 - my uid.

Ethical approval



UNIVERSITY OF ST ANDREWS
TEACHING AND RESEARCH ETHICS COMMITTEE (UTREC)
SCHOOL OF COMPUTER SCIENCE
ARTIFACT EVALUATION FORM

Title of project

Securing Digital Legacies

Name of researcher(s)

Josh Muir

Name of supervisor

Tristan Henderson

Self audit has been conducted [✓ **YES**] **NO**

This project is covered by the ethical application CS15727.

Signature Student or Researcher

Josh Muir

Print Name

Josh Muir

Date

20/09/2023

Signature Lead Researcher or Supervisor

[illegible]

Print Name

TRISTAN HENDERSON

Date

25/9/2023

A.3 User study method

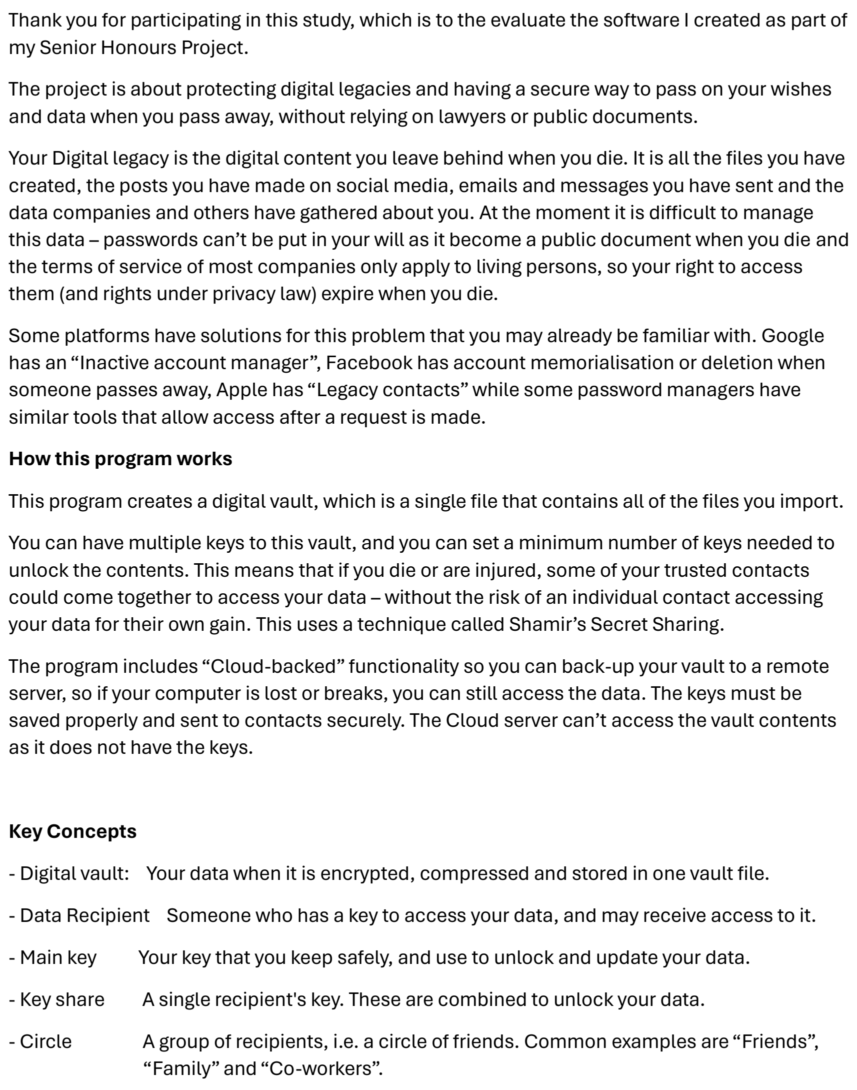
The user study was conducted to evaluate the usability of the application and determine how well it met the objectives. 11 participants took part, and they were students from the University of St Andrews across many schools - providing a range of technical and non-technical users. Participants were first given a sheet which included the required ethics statement, and then one with an introduction statement that provided a primer on the key terms and purpose of the application, and this is analogous to the basic knowledge someone may pick up from the website a program. Figure 25 shows the concepts page.

Participants where then given one of four scenarios:

- Scenario A - Creating an offline vault
- Scenario B - Unlocking an offline vault
- Scenario C - Creating a cloud-backed vault
- Scenario D - Unlocking a cloud backed vault

Scenario A and C are creation scenarios while Scenario B and D are unlocking. To prevent skill transfer effects the order in which participants completed these was randomised - i.e. Some did a creation task first, and others an unlocking task.

When a participant created a vault, they were asked to demonstrate opening the vault (with their main key) and asked what they would do with the keys. Once both scenarios were completed, participants a survey rating their knowledge of legacy, digital legacy and comments on the application. I emphasised to participants that the survey was anonymous and left them alone to complete it in an effort to remove any pressure or bias towards good feedback caused by the participants being existing friends of mine.



Thank you for participating in this study, which is to the evaluate the software I created as part of my Senior Honours Project.

The project is about protecting digital legacies and having a secure way to pass on your wishes and data when you pass away, without relying on lawyers or public documents.

Your Digital legacy is the digital content you leave behind when you die. It is all the files you have created, the posts you have made on social media, emails and messages you have sent and the data companies and others have gathered about you. At the moment it is difficult to manage this data – passwords can't be put in your will as it become a public document when you die and the terms of service of most companies only apply to living persons, so your right to access them (and rights under privacy law) expire when you die.

Some platforms have solutions for this problem that you may already be familiar with. Google has an "Inactive account manager", Facebook has account memorialisation or deletion when someone passes away, Apple has "Legacy contacts" while some password managers have similar tools that allow access after a request is made.

How this program works

This program creates a digital vault, which is a single file that contains all of the files you import.

You can have multiple keys to this vault, and you can set a minimum number of keys needed to unlock the contents. This means that if you die or are injured, some of your trusted contacts could come together to access your data – without the risk of an individual contact accessing your data for their own gain. This uses a technique called Shamir's Secret Sharing.

The program includes "Cloud-backed" functionality so you can back-up your vault to a remote server, so if your computer is lost or breaks, you can still access the data. The keys must be saved properly and sent to contacts securely. The Cloud server can't access the vault contents as it does not have the keys.

Key Concepts

- Digital vault:	Your data when it is encrypted, compressed and stored in one vault file.
- Data Recipient	Someone who has a key to access your data, and may receive access to it.
- Main key	Your key that you keep safely, and use to unlock and update your data.
- Key share	A single recipient's key. These are combined to unlock your data.
- Circle	A group of recipients, i.e. a circle of friends. Common examples are "Friends", "Family" and "Co-workers".

Figure 25: Concepts and explainer page of the participant briefing

A.4 Data Protection Impact Assessment

Step 1: Identify the need for a DPIA

Explain broadly what project aims to achieve and what type of processing it involves. You may find it helpful to refer or link to other documents, such as a project proposal. Summarise why you identified the need for a DPIA.
The project aims to serve as a “digital will”: A way of protecting the digital legacy of the deceased, and allow for their chosen contacts to access their data in a secure, privacy respecting way. The data may include very sensitive data including passwords, identity documents and their last will and testament. This means the project is both working with sensitive data of a highly personal nature and is an innovative technological solution, so a Data Protection Impact Assessment is required.

Step 2: Describe the processing

Describe the nature of the processing: how will you collect, use, store and delete data? What is the source of the data? Will you be sharing data with anyone? You might find it useful to refer to a flow diagram or other way of describing data flows. What types of processing identified as likely high risk are involved?

Data will be collected as the user manually submits files and folders to the project through the file import interface. Data sources include files on their computer and supported other integrations such as password managers, browsers or social media accounts.

This data will be stored on their computer and will never by transferred off the computer in an unencrypted form. The data will be encrypted according to specifications set by the user so that only they or the data recipients with whom they choose to share the keys can access it.

The data itself will not be processed beyond encrypting, transferring and decrypting it for user use.

The encrypted data may, if configured by the user, be stored on a remote server that is part of the project. Due to the nature of the encryption, the only information accessible by the server and project will be the user's name, email address and notification preferences.

Describe the scope of the processing: what is the nature of the data, and does it include special category or criminal offence data? How much data will you be collecting and using? How often? How long will you keep it? How many individuals are affected? What geographical area does it cover?
The nature of the data will be decided by the user but is broadly expected to be made up of identify documents, bill information, wills, account login information and other documents or images. It may include special category data. The unencrypted data consists of the user’s name and email address, which will be held indefinitely until they delete or access their encrypted data, and it will be stored with their encrypted data. Data will be collected whenever a user submits a digital vault to the online storage service.

Describe the context of the processing: what is the nature of your relationship with the individuals? How much control will they have? Would they expect you to use their data in this way? Do they include children or other vulnerable groups? Are there prior concerns over this type of processing or security flaws? Is it novel in any way? What is the current state of technology in this area? Are there any current issues of public concern that you should factor in? Are you signed up to any approved code of conduct or certification scheme (once any have been approved)?
The relationship is of software/service provider. The user has ultimate control over whether the data remains stored, the nature and content of the data, who has access to the data (through the key sharing mechanism), and whether it is stored remotely at all. The user is provided with explicit information on how the data is stored, and what parts of the data are stored unencrypted (name, email address), and that the rest of the data is stored in an encrypted form where only they have the keys. They do not include children or vulnerable groups. The underlying encryption and sharing mechanism is not novel, but the combination for the purpose of storing and sharing user data and wills may be novel. The same mechanism is used to share valuable secrets in other domains, such as cryptocurrency wallet keys.

Describe the purposes of the processing: what do you want to achieve? What is the intended effect on individuals? What are the benefits of the processing – for you, and more broadly?
Processing is minimised to respect user privacy. Processing is performed in order to provide user data with more resilience and security from unwanted data access – through unlock alerts and providing a long-ter storage solution. The intended effect is that extra vault functionality is possible: Unlock notifications and update reminders. There are no benefits for having access to this data, but users benefit from extra data security and resilience to data loss.

Step 3: Consultation process

Consider how to consult with relevant stakeholders: describe when and how you will seek individuals' views – or justify why it's not appropriate to do so. Who else do you need to involve within your organisation? Do you need to ask your processors to assist? Do you plan to consult information security experts, or any other experts?
Individuals views on this processing will be gathered using user studies on the software itself. Wider consultations with stakeholders are not possible due to project time constraints and not required due to the use of zero-knowledge encryption.

Step 4: Assess necessity and proportionality

Describe compliance and proportionality measures, in particular: what is your lawful basis for processing? Does the processing actually achieve your purpose? Is there another way to achieve the same outcome? How will you prevent function creep? How will you ensure data quality and data minimisation? What information will you give individuals? How will you help to support their rights? What measures do you take to ensure processors comply? How do you safeguard any international transfers?
Lawful basis: Consent. The lawful basis for processing is consent – the user will have explicitly consenting to the sharing and storage of the data by submitting it to the vault. Given the data may include special category data, explicit consent is required under Article 9 of the GDPR. Other ways to provide the functionality exist, in the form of current wills, but this project is intended to solve limitations of paper-based wills and keeping the data that is to be passed on private – so they do not give the same outcome. The data stored in an unencrypted form is minimised – only a name and email address are stored for the purpose of identifying the data subject, and carrying out the online functionality (i.e. notifications, unlock alerts). Users/Data subjects are responsible for maintaining the quality of the data within the 'vault'/encrypted section as it is not possible to inspect this data. Which information is encrypted and which is not will be explained to the user above the relevant fields, to make it clear what is only accessible to them and their recipients and what is not. To help support the data subject's rights, and that processors comply, unencrypted vaults will have a notice added explaining their obligations and to respect the subject's wishes. The use of unlock notifications and encryption mean it will only be possible for the recipients to access the data where the original subject is dead, and cannot make their wishes clear. To get explicit consent, the user will be asked for consent in the 'review' page of the application which explains clearly that the data will be encrypted and stored locally or uploaded, depending on the configuration settings they have chosen. Users can exercise their right to deletion (withdraw consent) by opening a cloud-backed vault and then clicking the delete button – or by deleting the vault file.

Step 5: Identify and assess risks

Describe source of risk and nature of potential impact on individuals. Include associated compliance and corporate risks as necessary.	**Likelihood of harm**	**Severity of harm**	**Overall risk**
Inability to exercise rights – Data Subject likely deceased when accessed.	Minimal	Severe	Low
Loss of control over use	Reasonable	severe	Medium
Identity theft or fraud	Likely	Severe	High
Loss of confidentiality	Reasonable	Medium	Medium

Step 6: Identify measures to reduce risk

Identify additional measures you could take to reduce or eliminate risks identified as medium or high risk in step 5				
Risk	**Options to reduce or eliminate risk**	**Effect on risk**	**Residual risk**	**Measure approved**
Loss of control	• Unlock notification and approval	Reduced	Low	Yes
Identity theft or fraud	• Unlock notification and approval • Guidance to recipients on how to use • Encryption prevents access without proper keys.	Reduced	Low	Yes
Loss of confidentiality	• Encryption prevents access • Multiple levels of trust means those with a 'low level' of trust cannot cooperate to gain access • Unlock notification and approval	Reduced	Low	Yes

Glossary

Alice Alice is a persona, and one of Bob's data recipients. 4, 10

Authenticated Cryptography Authenticated cryptography is an algorithm which encrypts one portion of data while signing another - it provides confidentiality for the ciphertext, and integrity for the ciphertext and extra plaintext data - known as Additional Authenticated Data. 22

Bob Bob is a persona, and the data subject. His data is in the system. 4, 10, 14, 15, 17, 61, 62

Carol Carol is a persona, and one of Bob's data recipients. 4, 10

circle A circle, as defined in the best practices draft paper [34] refers to a social group of data recipients or key holders, such as friends, family or colleagues. A circle can be set as required so that at least one of those keyholders is required to unlock the data. This means, for example, that family keys could be given the highest importance by making them required. 21, 26

cloud-backed vault A cloud-backed vault is a digital vault which is stored on the cloud provider. It can be accessed using either the main key or by combining the key pieces - without needing to provide a copy of the vault file. 16, 21

Dan Dan is a persona, and one of Bob's data recipients. 4, 10

data recipient "means a natural or legal person, public authority, agency or another body, to which the personal data are disclosed, whether a third party or not." [13, Article 4.1.9]. In the context of this project, a data recipient is someone who receives an encrypted copy of the data. They are a 'keypiece holder'. This is referred to by Facebook and some other platforms as a 'legacy contact'. 4, 6, 7, 9, 10, 14, 15, 17, 28, 30, 32, 37, 61, 62

data subject "An identified or identifiable natural person" [13, Article 4.1.7]. In the context of this project, Bob is a data subject. A data subject is someone whose data is being encrypted and shared. 4, 6–10, 14, 16–18, 23, 26, 27, 36, 37, 61, 62

digital asset "any electronic asset of personal or economical value" [1]. 4, 61

digital legacy Digital legacy refers to all of the digital assets belonging to a person when they die. It may also include their messages, profiles and other data resulting from online activity. . 6, 7

digital vault A digital vault refers to the personal data belonging to a data subject and the associated metadata. At a low level, it is a tar archive containing their files and all configuration which is then encrypted and prepended with some public information. 61

Erin Erin is a persona, and one of Bob's data recipients. 4, 10

Frank Frank is a persona, and one of Bob's data recipients. 4, 10

key piece A key piece is a piece of a combined key. It is a piece as defined by Shamir [20] and is sometimes also referred to as a keyshare. 9, 10, 14–17, 21–23, 26, 30, 36, 37, 43, 61, 62

key-share Other term used to describe a key piece in some literature. 26

main key The main key is the key for the data subject. It is used to encrypt the personal data and open/update the data as required. 17, 23, 30

natural person A person who is alive. Once a person dies, they cease to be a natural person. A natural person refers to a human only - not a company or otherwise legal person. . 61

personal data "means any information relating to a . . . data subject" [13, Article 4.1.7]. 6, 7, 10, 14, 15, 17, 37, 61, 62

post-mortem privacy Defined as "the right of a person to preserve and control what becomes of his or her reputation, dignity, integrity, secrets, or memory after death." by Harbinja [15]. 8

special category data "personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation" [13, Article 9.1]). 18

unlock notification An email sent to the data subject's email to indicate that an unlock request has been made. A grace period, defined by the data subject then takes place, and the data is released once it has elapsed. This only applies to cloud-backed vaults. 14

zero knowledge storage Zero knowledge storage is where data is encrypted locally on a user device and then uploaded in encrypted form to a storage provider. It has the advantage that the storage provider cannot inspect the data or disclose it to external parties as they never possess the decryption key [45]. 14