Spring 1-1-2025

Digital Identity Infrastructure

Jonathan Evan Hochman
Yale University Graduate School of Arts and Sciences, JEHOCHMAN@GMAIL.COM

Abstract

Digital Identity Infrastructure

Jonathan Evan Hochman

2025

A uniform infrastructure for identifying users and data on the Internet is proposed that would allow users to login to the Internet and then reveal as much or as little identifying information as they like to each digital platform they visit. Exemplary protocols are provided to demonstrate user authentication, federated identity, linked identity, privacy-preserving record linking, Sybil account detection, key recovery, and lawful access. A cryptographic method for privacy-preserving record linking (PPRL) supports the processing of medical research data. A novel extension of the RSA cryptosystem is described that allows two independently generated private keys to pair with a single group public key. This supports key recovery and lawful access methods.

Digital Identity Infrastructure

A Dissertation

Presented to the Faculty of the Graduate School

Of

Yale University

In Candidacy for the Degree of

Doctor of Philosophy

by

Jonathan Evan Hochman

Dissertation Director: Michael J. Fischer

May 2025

Contents

Acknowledgements

Joan Feigenbaum, Ben Fisch, Ruzica Piskac, Bruce Schneier, Aikaterina Sotiraki and Ted Wittenstein provided extremely helpful advice during the development of this work. My advisor, Michael J. Fischer, consumed several hundred cappuccinos while listening to my ideas, many of them at Katalina's Bakery in New Haven. Mike has been my mentor since 1986. His prior student Eva Syta spent many hours helping me and provided valuable feedback in the early stages of this work. Carl Pomerance patiently answered my questions about factoring algorithms and provided encouragement. Jessie Stricchiola first suggested the term "digital platform."

The UNS Project team has already implemented several of the protocols developed herein. Sophia Paleologou (another Fischer student) supervises software engineering and made many helpful suggestions. I also had many conversations with my co-inventors Jonah Stein and John Cunningham about the best ways to implement federated identity. Alex Todorovic manages the business. His twin sister Vanya Todorovic provided significant help until she succumbed to cancer at age 55. (This happened 191 years after Sophie Germain, the number theorist whose discoveries serve as a foundation of this work, also died at age 55 from the same disease.)

A major goal of this dissertation is to facilitate research based on medical databases that could lead to better cancer treatments. Daniel Boffa, Chief of Thoracic Surgery at the Yale School of Medicine, has had numerous discussions with Dr. Fischer and me over the last five years about the need for privacy-preserving record linking. Dr. Boffa is leading

an effort to deploy a medical research database system throughout the United States based on the protocols described in Chapter 11. We have also had multiple conversations with Cynthia Dwork and her graduate students about using differential privacy to mitigate the risk that deidentified medical records might be re-identified.

Starting in 2008 I had a series of conversations with Matt Cutts about Internet security and federated identity. Mr. Cutts was the head of the web spam team at Google and later served as Director of the United States Digital Service.

In 1987 Stanley Eisenstat arranged a summer job for me to program a remote user authentication protocol based on trusted hardware. It required monkeying around in the session layer with sockets and ports.

In 1984 Alan Perlis spent 90 minutes with 16-year-old me and changed my path from physics to computer science.

List of Figures

Preface

This dissertation proposes a uniform infrastructure for identifying users and data on the Internet. It identifies the questions we need to ask, provides terminology to think and speak about digital identity, introduces new cryptographic primitives to support digital identity services, and finally presents a sequence of applications that could be built upon the foundation of a shared digital identity infrastructure.

These solution sketches are not meant as final specifications. They are examples to demonstrate how the digital identity infrastructure could make it easier for digital platforms to deploy new services, improve privacy, reduce user frustration, and support the rule of law and civil society.

Chapter 1 starts with a statement of the problem to be addressed. Chapter 4 provides a lexicon. We need specific words and definitions to think clearly about digital identity problems. Chapter 3 explains the two main engineering challenges that must be addressed in the design of a digital identity infrastructure. Chapter 4 provides philosophical arguments to establish a set of desired properties called Just Enough Trust which serve as guidelines for designing identity systems. Chapter 5 addresses legal issues that should be considered when designing an identity system.

Chapter 6 introduces the concept of digitally signed tickets as a building block of identity transactions that satisfy the requirements of Just Enough Trust. Chapter 7 sketches a basic identity system supported by tickets. Chapter 8 introduces the concept of trusted agents and explains why they are useful. Chapter 9 shows how to use tickets and trusted

agents to build a federated identity system on an Internet scale. Chapter 10 extends this system to provide a link between online identities and legal identities in a way that preserves privacy.

Chapter 11 introduces a novel blinding-completion cryptographic method designed to support a system for managing private data, including a feature called privacy-preserving record linking (PPRL). PPRL is of great interest to the medical research community and has many other potential applications. The explanation of blinding-completion functions delves into discrete mathematics. Chapter 12 explains how to combine linked identities with PPRL to produce a Sybil account detection system.

Chapter 13 introduces the group public key (GPK) cryptographic method, a way to have two distinct cryptographic keys that can open one lock. This method is extremely useful for building resilient systems that can help users recover after losing their keys. The GPK method also provides the basic elements required for building a system of lawful access by which the rule of law can be extended to digital platforms. Readers who do not love discrete mathematics can jump over the theorems and proofs in this chapter.

Chapter 14 sketches a key recovery system that doubles as a method of lawful access. Chapter 15 concludes with public policy recommendations to accompany the innovations of the previous chapters.

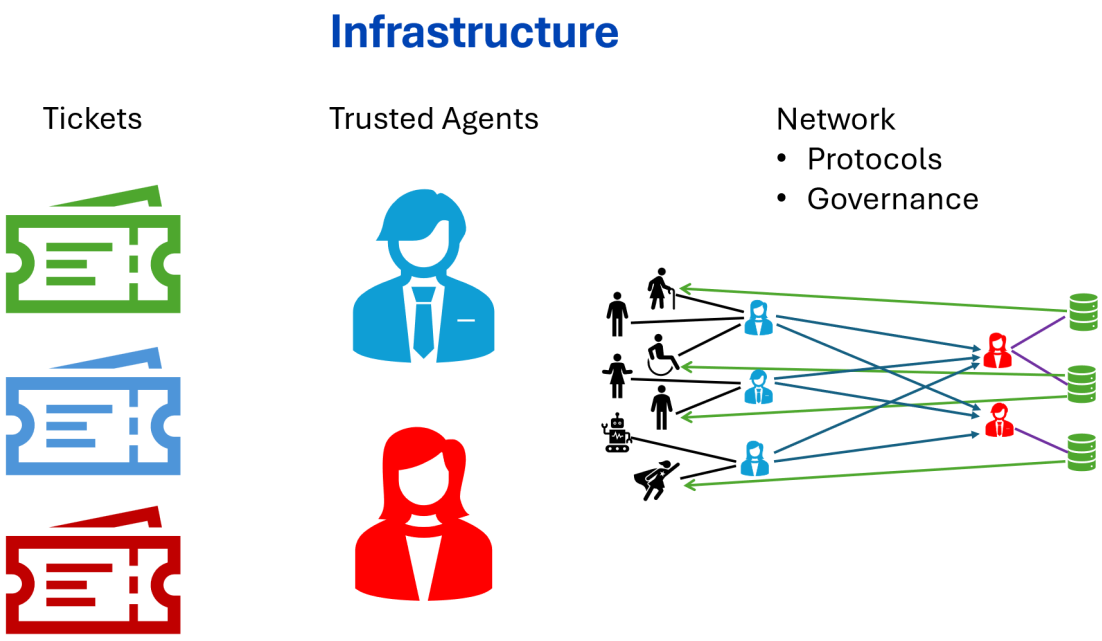

Figure 1: Digital Identity Infrastructure

Chapter 1

Introduction

The Internet has become insufferable. Current digital identity systems utilize an ad hoc collection of password-based and multi-factor authentication methods. The inconsistency of these methods leads to user confusion, inconvenience, lack of interoperability, poor privacy properties, and degraded security. Data breaches, account takeovers, and privacy violations have become the norm. Efforts to combat these abuses have only been marginally effective. To stop account takeovers, users are faced with frustrating login requirements where they have to prove who they are over and over again. Fake accounts still cheat online advertisers, skew ratings, and stir up divisions on social media by amplifying disinformation. The Internet promised a new era of free knowledge, convenience, and productivity, but these goals have been undermined by a profusion of complicated, ineffective, and inconsistent digital identity systems. As shown in Figure 1.1, these multiple identity systems often cohabit on the same login page creating too much complexity. In addition, existing online identity systems lack sufficient features to effectively deal with messy edge cases. What if we need to log into our spouse's account, with consent, to check and pay a credit card balance? What if two spouses want to share an account to order pet food online? What if parents want to monitor the online activities of minor children? What if a legally appointed guardian needs to supervise the activities of an incompetent adult? Why do online services

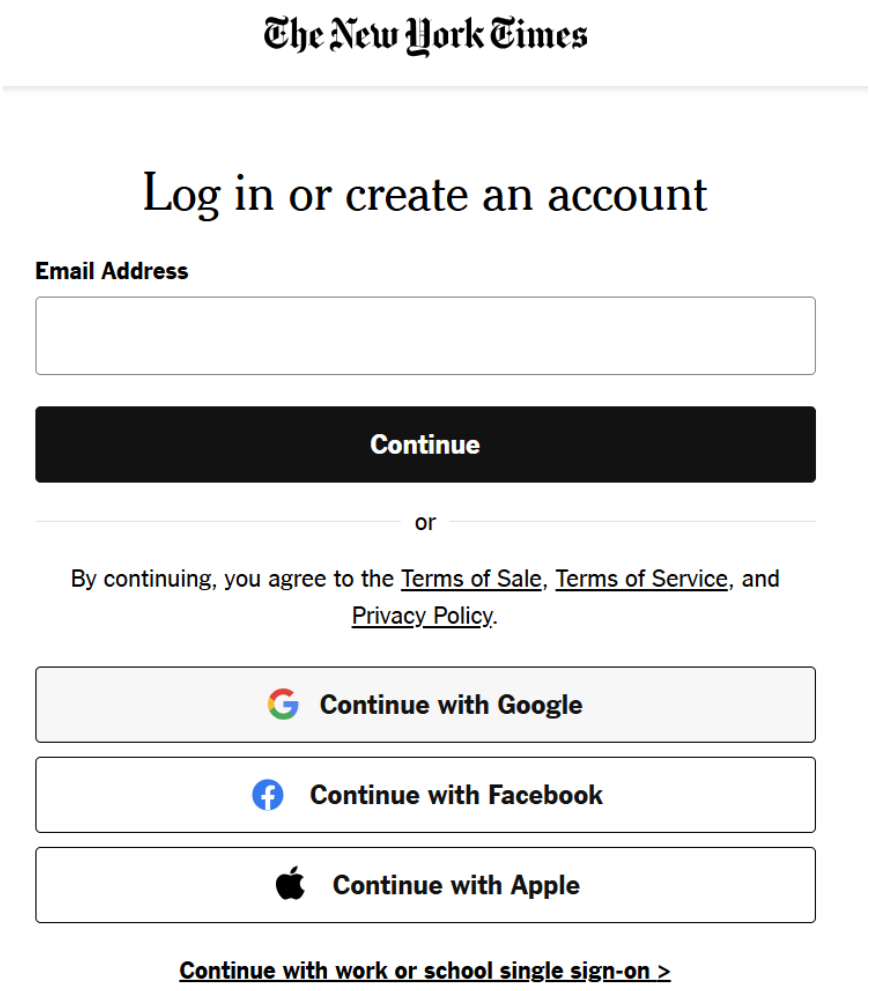

Figure 1.1: A typical login screen at a popular digital platform [5].

know so much about our private browsing, yet medical researchers struggle to fully utilize data from our past treatments? What if the government has a search warrant and legitimately needs access to a suspect's online accounts? What if someone dies and their next of kin needs to take control of online assets?

Existing *ad hoc* identity systems were not conceived with the goal of supporting the rule of law. Often they do the opposite because they allow for the creation of lawless online spaces. Such spaces are fashionable with the technocratic elites who have embraced libertarianism, where the individual is entitled to do anything they want, free of government interference. With the Internet central to daily life, this cannot continue if we wish to maintain the rule of law and democracy.

Historian Heather Cox Richardson explained, quoting Abraham Lincoln's Lyceum address, "A public practice of ignoring the law eventually broke down all the guardrails designed to protect individuals, while lawbreakers, going unpunished, became convinced they were entitled to act without restraint. 'Having ever regarded Government as their deadliest bane,' Lincoln said, 'they make a jubilee of the suspension of its operations; and pray for nothing so much as its total annihilation.' The only way to guard against such destruction, Lincoln said, was to protect the rule of law on which the country was founded...Lincoln was quick to clarify that he was not saying all laws were good. Indeed, he said, bad laws should be challenged and repealed. But the underlying structure of the rule of law, based in the Constitution, could not be abandoned without losing democracy" [81].

Just because we do not like the law (or some laws), we are not free to thumb our noses at it. Our digital identity infrastructure must be designed to work with and support the rule of law. If these laws are defective, we have to do the hard work of lobbying to change them.

The commercial Internet and later the World Wide Web evolved incrementally over decades with developments focused on solving specific problems that would allow the operators of digital platforms to amass fortunes. Internet identity is a hard problem that has eluded a good solution for decades, but we seem to have reached a point where people are

ready for change, even if they are unsure what that change looks like. Design is inherently a philosophical endeavor that requires prioritizing values. What is good? What is bad? With these value judgments in hand, it becomes possible to engineer a system that achieves the desired properties.

To change the Internet from what it is to what it could be, we need to implement policies that encourage technologies designed to benefit people. We need technologies that support privacy, give truth a fair chance to prevail over lies, and resist the formation of lawless online realms. Our challenge is to devise a system that can accommodate both the right to privacy and the rule of law.

The solutions to these problems would require considerable investment and a high degree of duplicate effort if undertaken independently by each digital platform. The situation is ripe for an infrastructure approach, the same way societies have worked together to build seaports, the electric grid, water treatment plants, and the national highway system.

What would some of these ameliorative technologies look like? Rather than logging into each site, users should be able to log into the Internet, and then be able to conveniently share as much or as little information as necessary to prove that they are authorized to access information and perform transactions.

Currently, identity is handled in the Internet application layer, layer 7 of the OSI Model [45]. This leads to enormous duplication of effort and unnecessary complexity because each application needs methods to identify users. Instead, we could turn identity into an infrastructure service by pushing it down to the presentation and session layers (layers 6 and 5). Applications could then ask the network service: "By what name do I know the user on this connection?"

Not only should users be able to control their identity, it should be possible to tag data (including deidentified data) for easy processing by different sites. Our doctors should be as good at predicting which treatments would save our lives as Amazon is at predicting which products we want to buy. Using artificial intelligence to make such predictions is

straightforward once the data become available, but current systems severely hinder the use of deidentified medical data for research.

Procedures for verifying user information should support legal compliance, such as enforcing age and citizenship requirements, or checking whether a pseudonymous user is on a sanctions list. The fundamental problem of disinformation is that liars have no inhibitions about creating an army of fake accounts to subvert the algorithms that decide which content to display, rank, or amplify. Those who speak the truth tend to follow the essential rule of democracy: one person, one vote. To give the truth a chance to prevail over lies, we must have a way to stop fake accounts from abusing digital platforms.

At minimum, a comprehensive identity infrastructure would include the following services:

1. Basic identity, where users can identify themselves consistently whenever they connect to a remote site (Chapter 7),

2. Federated identity, where users can login once and then identify themselves with a unique pseudonym at each site they visit (Chapter 9),

3. Linked identities, where each pseudonym is attached to a real life identity that is kept private, allowing users to verify facts about themselves and prove legal compliance (Chapter 10).

4. Privacy-preserving record linking, which allows deidentified data fragments to be tagged with pseudonyms in a way that allows fragments related to a single entity to be linked without identifying that entity (Chapter 11).

5. Sybil (fake) account detection, to stop abuse of on-line platforms and the amplification of disinformation (Chapter 12).

6. Key recovery and lawful access, to allow online activity to become more reliable, resilient, and subject to the rule of law (Chapter 14).

Chapter 2

Lexicon

To have a thoughtful discussion about Internet identity, we need words to describe the concepts we will use. Many of these words have common meanings in the fields of computer security or cryptography under the assumption that people know what they mean. Here, these words will be defined with greater specificity.

2.1 Players

An Internet identity system has a variety of **players**, independent actors who pursue their own agendas. The players could be natural people, organizations, or automatons. For example, each Googlebot is a player because it interacts with websites via the HTTP(S) protocol.

One type of player is a **digital platform, Internet platform, or platform**, a tightly integrated set of computing systems and software connected to a network that provides data services to remote users and allows them to do interesting things. A social network app such as Instagram is a platform. The Instagram app together with the Instagram website and other Meta apps that all share user information are together a platform. Amazon is a platform. Bank of America's online banking services are a platform. The Canvas e-learning system is a platform. The New York Times website is a platform, even for users who do not

have an account, because the website issues cookies, assigns identifiers, and tracks usage by anonymous users.

A **user** interacts with digital platforms over the Internet or locally at a terminal such as an automated teller machine (ATM). When a user interacts with a platform, their data is often stored in an **account** for convenience and to provide a persistent state for transactions between the user and the platform. A principle data security concern of platforms is to allow users to view their own data and not to view data of other users without proper consent. For example, with online banking, a user should ordinarily only be able to access their own account balances.

An **agent** is an internet-connected player that acts on behalf of other players to automate or facilitate specific types of transactions. For example, a platform could also act as an agent for users or other platforms.

2.2 Identity

At the most fundamental level, the word **identity** means ”the same.” When we identify an object, we are making an assertion that the object is a specific one, and not some other one. Identity answers the question, is the object in one reference the same object as that in another reference [79]. For example, the sentence ”I am Jonathan Hochman” asserts that the object referenced by the word ”I” is the same one as referenced by the proper name ”Jonathan Hochman.”

Identification is the process of associating an **identifier** to an object and is important in computer systems. An **object** is some type of person or thing that we want to track. In a computer system, things are represented as data records or objects. As Grace Hopper noted in 1982, each data record needs to have an identifier so that you can tell them apart [23].

Identity should not be conflated with **attributes**. For example, attributes of an object such as first name, date of birth, sex, hair color, or DNA code are not identity, although

they may be an aid in the identification process. A set of attributes can be attached to an identity via data records. Attributes or a set of attributes can sometimes be used to identify an object, especially if the attributes are specific enough to match only one entity.

There are different types of identifier. A **name** is a canonical identifier for an object. Names could be proper names, unique numbers, or any other unique token. An **alias** or **pseudonym** is a non-canonical label that identifies an object. Any unique attribute, such as a social security number, could also be used as an identifier.

Anonymous means without an identifier. An anonymous object could still be identifiable on the basis of its attributes. **Pseudonymous** means having a pseudonym, a name different from an object's canonical name, such as the one that appears in legal identity documents like a passport or driver's license.

Deidentified means a data record with its personally identifiable information (PII) removed. For example, a medical record when the patient's name and social security number have been removed may be considered deidentified. **Re-identification** is a process whereby the attributes of a data record can be used to infer the subject's personally identifiable information, possibly with reference to outside data sources.

A **nonce** is an ephemeral identifier, a number that is used only for a single transaction. It is a throwaway name, often generated randomly in a large name space such that the probability of nonce reuse is negligible. Nonces are frequently used in network and security protocols [67].

A **hapax legomenon**, shortened to **hapax**, is an ancient term in linguistics and biblical studies that means "once said" [44]. A hapax is a word that appears only once in a corpus, a body of words. Thus, if a corpus is the set of identifiers used across the entire Internet and a hapax is assigned as an identifier on one digital platform, that means no other platform will use the same identifier. As will be explained, the use of hapaxes as identifiers is an important measure for preserving privacy. Hapaxes are similar to universally unique identifiers (UUID) and globally unique identifiers (GUID) in terms of function, although they

are generated with a different protocol [69]. In effect, hapaxes are a generalized version of UUIDs and GUIDS. Unlike nonces, hapaxes are persistent identifiers.

A **Sybil account** or **Sybil** is an alternate identity of a user who has another account on a platform, and the relationship between the accounts is undisclosed to the platform. For example, if someone figured out how to cheat Bank of America and created 100 different accounts on bankofamerica.com to rip off the bank 100 times, those would be Sybil accounts. If someone created 100 Amazon accounts to leave glowing reviews for a product they were selling on Amazon.com, to cause their product to rank higher on the Amazon search and to better persuade prospective customers, that would be called **review stacking**, which is a type of **Sybil attack**. When Vladimir Putin directs his minions to create tens of thousands of fake social media accounts to spread disinformation and destabilize civil society around the world [87], those campaigns are Sybil attacks.

2.3 Privacy

Privacy is the right to be free from unwanted surveillance or intrusion into personal matters. Privacy is an internationally recognized human right [18]. A network protocol is **Privacy-preserving** when the participating players do not learn any attributes of the other players beyond what data each player chooses to share. For example, when a user logs into a digital platform with privacy-preserving properties, it is not necessary to provide an email address or mobile phone number as part of the user authentication process. These attributes are highly identifying and could be used to link data records from other platforms to build a detailed dossier of user's activities across the Internet.

A closely related concept to privacy is **confidential information**, which means information whose distribution is restricted to authorized parties only. When exposure of confidential information would risk harm to someone, it is also called **sensitive information**. It can be difficult to tell when confidential information is sensitive because circumstances

can vary from person to person.

In the context of a privacy-preserving system, user information belongs to the user and may not be shared without the user's agreement. This requirement comes into play when designing an identity system because poorly designed identifiers can leak information.

2.4 Trust

Trust means to place confidence in another party to carry out their duties properly. Trust can be broken due to incompetence or malice. The party placing trust exposes themselves to the risk of injury should their trust be breached. On the other hand, a trusted party assumes the risk of being falsely accused of breaching trust.

A **trusted agent** is a professional data service provider. Trusted agents may hold confidential or sensitive information on behalf of users and promise to loyally protect the interests of their users.

Confidence, in the context of trust, is the belief that a party will carry out their duties properly. Confidence in a party may develop for a variety of reasons.

- Having a reputation for honesty.
- Being perceived as competent.
- Having values consistent with the trusting party.
- Having the financial resources, or a third-party assurance, that guarantees the ability to compensate the trusting party if something goes wrong.
- Maintaining membership in a professional association that screens its members.
- Providing a service level guarantee.
- Being subject to government regulation and oversight.

- Being free of actual or perceived conflicts of interest.

Confidence could also develop through the tactics used by swindlers and con men.

- False advertising, puffery, or bragging.
- Defaming competitors.
- Using Sibyls to influence public opinion.
- Employing psychological manipulation.

2.5 Cryptography and Computer Security

A **protected secret** is a token or number that has not been shared and is stored in a secure enclave. A **secure enclave** is a hardware subsystem for storing secrets that performs limited calculations and outputs based on the secrets. A secure enclave is physically incapable of revealing the secrets it contains. Modern computing devices, including mobile phones, laptops, and desktop computers, have secure enclaves. Protected secrets cannot be copied, backed up, or synced to other devices.

A **secret key** is a secret that can be used as part of a cryptographic protocol that implements functions such as encryption, decryption, or digital signatures. A **public key** is a key value used in a cryptographic protocol that can be transmitted and copied. This is an unfortunate misnomer because public keys are not necessarily publicized. Thus, a public key may or may not be widely shared. We will see that privacy-preserving protocols may require public keys to be kept confidential because public keys can be used to identify a player. To help reduce confusion, this dissertation will often refer to public keys as **verification keys** when such keys are used to verify digital signatures. This nomenclature is more precise and less susceptible to confusion.

Authentication is a protocol, usually based on cryptography, that is used to verify identity or facts. A closely related concept, **authorization**, is a protocol by which a player

consents to a transaction, such as retrieving confidential information or issuing an online payment. An important type of authorization is pinning an identity commitment, as will be explained below. When a player pins an identity commitment, they consent to be legally bound by digital signatures that verify with the identity commitment.

An **identity commitment** is a unique value or ticket that is mathematically associated with a secret. Although the secret cannot be shared, the identity commitment can be shared without revealing the secret. The identity commitment allows the receiver to generate a **puzzle** that can only be solved by a player who knows the secret. In chapter 7, an example of such an identity commitment scheme will be detailed. Identity commitments are useful for building user authentication protocols. The user can **pin** their identity commitment with a platform upon establishing an account. Later, when a player wants to login, the platform can challenge the player to solve a puzzle based on the pinned identity commitment and thereby prove that they are the same user who provided the commitment.

A **data breach** is a type of computer security failure that reveals confidential information to an unauthorized player or the public. A **catastrophic breach** of a system is one in which substantially the entire data set of the system could be revealed to an unauthorized player or the public. More broadly, a **catastrophic failure** is a failure that causes the system to collapse and stop processing transactions.

Probabilistic algorithms, which are algorithms that use a random input, are ubiquitous in cryptography and computer security. Consequently, when a definition anywhere within this dissertation incorporates a categorical modifier, such as “all”, “always,” “none,” “never,” “unique,” “only,” or “only once,” there is a negligible probability of an exception. A negligible probability is one so small that it will not happen in a time frame meaningful to human history, such as the chance of flipping a fair coin 54 times in a row with each result being heads. At the rate of one coin flip per second, this should only occur once per 570 million years.

Formally, **non-negligible probability** means $p \geq \frac{1}{f(x)}$ as x becomes very large, where

f is a polynomial function and x is the security parameter of the protocol, such as the length of an encryption key. In contrast, **negligible probability** means $p < \frac{1}{f(x)}$ as the security parameter x becomes very large, for every f that is a polynomial function. The requirement that x must become very large means that these are only asymptotic properties.

2.6 Distributed Systems

A **distributed system** is a collection of platforms and users that may be in different locations, connected by communication links such as wired or wireless networks that carry messages. The theoretical model of a distributed system mirrors the real world and all its surprises. Any of the computers or software involved in the distributed system could crash, freeze, or go into an infinite loop, the communication channels might fail, and messages could be delayed for an arbitrarily long time, or be lost altogether.

In the context of a distributed system **consensus** means that some group of distributed computing processes agree on value. A classic example is the Byzantine Generals Problem. Several allied armies besiege an enemy. The allies will be victorious if they attack together, but will lose if they attack separately. To coordinate an attack, the only means of communication are messengers that can be sent between the allied armies, but the messengers must cross enemy territory where they could be captured. The problem is to develop a protocol to coordinate a simultaneous attack.

Chapter 3

Engineering Considerations

When building a distributed system for digital identity, there are numerous engineering considerations, most of which can be addressed through careful implementation. Two issues are fundamental and must be addressed in the design of the system. These issues are distributed consensus and isolation.

3.1 Distributed Consensus

In 1985, Fischer, Lynch and Paterson proved that it is impossible for any protocol to guarantee the formation of a distributed consensus in the presence of even one faulty process [58]. This surprising result became the fundamental theory of distributed systems and subsequently gave rise to a rich body of research on distributed consensus algorithms under a wide variety of conditions. A related area of commercial development that has become quite popular is blockchain and cryptocurrency.

The **Fischer problem** refers to this impossibility of ensuring distributed consensus in the presence of even one faulty process. Any system for remote identification over the Internet is subject to the Fischer problem because no system is immune to the potential for hardware, software, or communication failures. When the Fischer problem strikes, a distributed transaction might become **wedged**. A wedged transaction is one that fails to

complete and may leave the distributed system in a logically inconsistent state. In contrast, a **committed** transaction is one that successfully completes and leaves the distributed system in a logically consistent state.

To effectively address the Fischer problem, a distributed system must be able to recognize when synchronization has failed, and have an out-of-band process that does not rely on the distributed network's communication channels to resolve the disagreement. As explained in Chapter 8, one way to address the Fischer problem is to remove the distributed nature of the transaction by providing a way for players to meet face-to-face.

3.2 Isolation

The property of **isolation** means that two or more nodes in a distributed system operate independently of each other, although they are still able to communicate. The probability of one node failing is independent of the probability of the other nodes failing. Isolation is a powerful technique that can enhance security. In a well-designed system, isolation ensures that the failure of a single node does not cause a catastrophic failure of the entire system. Proper design avoids creating such single points of failure, where one thing failing causes a cascade of failures that wipe out the whole system.

One classic method to achieve isolation is an $N - K$ secret sharing scheme [85]. In such a scheme, a secret is divided into N pieces and stored with N independent nodes. The secret can then be recovered from any $K : K < N$ of the shares. The failure of $N - K$ or fewer nodes will not prevent the secret from being recovered. Likewise, if fewer than K nodes are subject to a data breach where their shares become exposed, the adversary will not have enough shares to reconstruct the secret.

When a system has been designed with isolation, at least two nodes need to fail within the time period Δ, the maximum time that the system takes to remediate a failure, to trigger a catastrophic failure. Depending on the nature of the system, remediating failures could

involve monitoring and restarting hardware, rotating cryptographic keys, or scanning and reinstalling software. With isolation, if the probability of a single failure is p, then the probability of a catastrophic failure is at most p^2.

By increasing the isolation parameter κ, the number of independent node failures within Δ required to trigger a catastrophic failure, a distributed system can be made more reliable. If the probability of failure of a single node within Δ is p, then the probability of catastrophic failure within Δ is p^κ. It becomes an engineering question to determine how much reliability is enough and whether extra reliability is worth the costs.

Chapter 4

Social Considerations

Social considerations are just as important as network and security engineering because they determine whether anyone will use the digital identity infrastructure. A system with excellent engineering properties that nobody uses serves little purpose beyond an esoteric thought experiment. To design a system that people will use, social and psychological requirements must be taken into account.

4.1 Just Enough Trust

Just Enough Trust is a set of philosophical statements about values. These values can guide the design of an Internet-scale identity system, one that will have a wide appeal to a global audience and will also support the rule of law. There are four major requirements for an Internet identity system to comply with Just Enough Trust:

1. Users can choose who they want to trust.

2. Users can detect and provide proof if a trusted agent breaches trust.

3. Trusted agents are safe from unfounded claims.

4. The system preserves privacy whenever possible.

In shorthand, we can label these requirements “freedom of choice,” “accountability”, “safety,” and “privacy.”

4.2 Freedom of Choice

Freedom of choice means that a player can choose who the player wants to trust. Each player can choose one or more trusted agents that the player prefers to act on their behalf. For example, if Alice is a user and Bob is a platform, both Alice and Bob can each choose a trusted agent to process identity transactions on their behalf. In Chapter 8 the significant benefits of using trusted agents will be explained.

Because Alice and Bob could choose different trusted agents, it is desirable to allow each party to a transaction to choose their own trusted agent. Nobody should be compelled to trust someone they do not want to trust. For instance, if Bob sets up his identity system to use Login with Facebook, but Alice does not trust Facebook, she should be able to choose a different identity provider, and Bob’s agent should be able to work with Alice’s agent to verify any proof of identity that Alice needs to provide to use Bob’s platform.

4.3 Accountability

Accountability means that if a player feels that a transaction has been performed involving them that they did not authorize, they should be able to challenge the validity of the transaction and the system should provide sufficient evidence to make a determination as to whether the transaction was properly authorized. The system should provide proof of proper authorization in the form of records that cannot be forged with more than negligible probability, even if the record keeper is dishonest.

If a player, Eve, breaches the trust of another player, Alice, then Alice should be able to detect that she has been cheated by Eve. Alice should also be able to gather evidence from the system to prove that she was cheated. Then Alice should be able to summon Eve

to participate in dispute resolution before a neutral adjudicator.

Accountability requires:

1. A breach of trust can be detected.

2. A breach of trust can be attributed to a specific player.

3. A breach of trust can be proven to the neutral adjudicator with evidence provided by the identity system.

4. A player accused of breaching trust can be compelled to participate in dispute resolution.

5. The system is free of conflicts of interest.

It must be possible to satisfy these requirements even if a player acts in bad faith, deletes data records in their possession, or makes false statements.

Existing user-authentication systems frequently lack the accountability property. For example, widely used password authentication relies on the user typing their password to access a system. The password is hashed by the platform and compared to a stored table of password hashes to verify that the user is who they claim to be. A major problem with such a system is that if the user denies that they logged in to perform a transaction, there is no way for the user to falsify the platform's claim that the login was properly authorized. The situation turns into a standoff where the user says "That was not me" and the platform says "Yes, it was." Because the platform possesses the table of password hashes, they are the ultimate authority on what password is accepted for that user.

One reason for using authentication protocols that rely on protected secrets is to establish accountability. If a secret has not been shared with other players, any proof based on that secret can be attributed to the player holding the protected secret. That player is therefore responsible for the transactions they authorize. They cannot point the finger at some other player because no other player has access to the protected secret. To make this

work, the relevant protocol or system needs to incorporate the rule that players must protect their secrets and that they may be held responsible for any transaction authorized with their secret.

Another problem with sharing secrets used for authentication is that protocols for sharing secrets require a method to authenticate the party receiving the secret. Even a protocol that allows for sharing a secret over an insecure communication channel requires that the player sending the secret must authenticate the party they are sharing with. Even though a protocol may protect against an eavesdropper listening to the channel, that does not help if an attacker impersonates the intended recipient. An authentication protocol should not rely on authentication because that would be a circular dependency. Therefore, authentication secrets should not be transmitted because there is no way to transmit them without already having a method of authentication. If such a pre-existing authentication method were available, there would be no need to create a new authentication method.

4.4 Safety

Safety is the property that an identity system can provide a player with sufficient evidence to defend themselves if they are falsely accused of breaching trust. Safety is the opposite side of accountability.

For example, if a platform authenticates a user, the platform could retain an admission ticket signed by the user as proof that the user authorized the user authentication transaction. Although this sounds simple, considerable effort and infrastructure may be required to keep track of the verification keys necessary to verify each user's signature.

Safety is also a problem for users. If a user is accused of logging into an on-line file sharing service and uploading contraband material, such as child sexual abuse material (CSAM), an innocent user needs to have a way to defend themselves if their account was hacked. The user should have a means to prove that the person logging in was not them.

Many existing systems lack effective defenses against a user who's account is compromised via a phishing attack or keystroke logger that steals their password, or if a password is reused at multiple platforms and one of the platforms is breached. The platform can say "That transaction is attributable to the user because the correct password was entered", and the user has no means to falsify the assertion.

4.5 Privacy

To protect privacy, an identity system must not reveal any information about the user other than what is necessary and what the user consents to share. For example, if a system attempts to perform an age verification transaction, asking the user to submit a photograph of a government issued ID would not be privacy-preserving if the ID contains additional information beyond age. The ID may include name, address, driver's license number, and other information that is irrelevant to the transaction and could put the user at risk.

Consider the hypothetical case of a woman who purchases a bottle of wine. An employee at the package store needs to check her driver's license to verify age, but that also discloses her home address. This could make her feel uncomfortable. People who have never been the target of harassment or stalking may not value their privacy. However, identity systems should be designed to protect everyone, not just the least vulnerable members of society. Sensitive information, such as home address, should not be revealed in the course of transactions that do not require such information.

Commonly used multi-factor authentication schemes require the user to provide an email address, mobile phone number, or both. Once the platform has that information, it can access commercial databases where the email address or mobile number serves as a lookup key. The platform can then download additional data about the user without the user's consent, or the platform could sell whatever data it acquires from the user to data brokers who add it to the user's dossiers. Privacy-preserving identity services would re-

duce the opportunities for these types of non-consensual data exploitation. When a user attempts to prove their identity, there is no reason to reveal sensitive information that is irrelevant to the transaction.

Returning with the package store example, one way to preserve privacy when doing an age verification transaction would be for the user to provide their government-issued identity document to a trusted agent. The trusted agent could then testify that it had seen the user's government issued identity document and could verify that the user is at least 21 years old. This transaction could be performed by a point-of-sale terminal and the user's mobile phone. The terminal could show the customer's picture, extracted from her government-issued ID, and her age. No other information is required.

Chapter 5

Legal Considerations

An important goal of digital identity infrastructure is to support the rule of law and especially to refrain from creating lawless realms. Supporting the rule of law means that the system should integrate and interface neatly with existing legal systems and practices. For example, the system should include a method for responding to court orders such as subpoenas for the production of data. This issue, lawful access, will be addressed in Chapter 14. The system should also provide a way to connect people with their digital property rights, an area of law that remains underdeveloped but is beginning to be addressed by legal scholars. [49]

5.1 Lawful Realms

A lawful realm is an area of cyberspace where disputes are resolved by applying reason, logic, and scientific methods to **evidence**. "Evidence [is] an item or information proffered to make the existence of a fact more or less probable" [16]. Over time, a great deal of effort has been spent thinking about what constitutes proper evidence. For example, in US federal courts, the Rules of Evidence have been codified and are frequently updated. [27]

The evidence is collected and shared during a pretrial process and then presented at trial to a neutral fact finder, either a judge, arbitrator, or a jury supervised by a judge, according

to established procedures based on due process of law. As with the rules of evidence, a great deal of effort has been spent addressing the question of what constitutes due process. What is fair? What is a reasonable amount of effort to invest in a dispute? How is a neutral adjudicator selected? Where is the right place to hold a trial? How are the jurors selected? Who has a right to trial by jury? How are conflicts of interest handled?

In US federal courts, there are two sets of rules: the Federal Rules of Civil Procedure [25] and the Federal Rules of Criminal Procedure [26]. In addition, each state has its own set of procedural rules, often modeled on federal rules [2]. Given the considerable investment in effort to make these rules, digital identity systems should be designed with an eye toward facilitating their operation rather than trying to reinvent them. We already have rules that spell out due process in great detail; we should reuse them, supplementing as necessary.

If a user has been accused of engaging in an illegal transaction, the social property of accountability means that the system should be able to provide sufficient evidence that the user can be held accountable. However, the social property of safety means that an innocent user should not be held liable for wrongdoing. The innocent user must be able to falsify an accusation, either by presenting their own evidence or by arguing that the claimant has failed to prove the claim with sufficient evidence. As discussed in Chapter 6, digital identity infrastructure can provide tickets as the basic unit of evidence.

Moreover, a lawful realm requires that on-line users can be connected to legal identities so that people can be summoned into a court of law to adjudicate a claim. There is little comfort in having evidence to prove a claim if the respondent cannot be summoned to court. Chapter 10 explains how that could be done with linked identities.

A **proof** is a rational argument based on logic and evidence. In a lawful realm disputes are resolved by a neutral judge or arbitrator after each disputant is given the opportunity to present their proof.

5.2 Lawless Realms

Lawless realms are areas of cyberspace where anything goes. An infamous example is the defunct ”Silk Road” website, which ”was used by more than 100,000 users to buy and sell hundreds of kilograms of drugs and other unlawful goods and services.” [12] The promise, often illusory, of anonymity was central to the Silk Road value proposition. According to the US Department of Justice, “Ross William Ulbricht operated Silk Road – a clandestine global marketplace that offered buyers and sellers of illegal goods and services a promise of anonymity. Ulbricht built this black market bazaar to exploit the dark web and the digital currency Bitcoin to allow users to conduct illegal business beyond the reach of law enforcement. Ulbricht’s arrest and conviction – and our seizure of millions of dollars of Silk Road Bitcoins – should send a clear message to anyone else attempting to operate an online criminal enterprise. The supposed anonymity of the dark web is not a protective shield from arrest and prosecution.” [12] Despite the effective prosecution of this case, the offer of anonymity was sufficiently enticing and sufficiently difficult to pierce, that the marketplace grew to enormous size before it could be shut down. In 2025, President Donald Trump pardoned Ulbricht, claiming that the sentence was government overreach. In his defense, Ulbricht claimed that his goal in creating the Silk Road was “to empower people to make choices in their lives and have privacy and anonymity” [34]. By creating the illusion that users were beyond the law, the Silk Road provided a venue where criminality thrived.

Although platforms should be able to offer private and anonymous services, such promises should be subject to the operation of law. User information can be kept private, and users can interact without providing legal names, but if they engage in criminal activities as defined by their local legal system, they run the risk of losing their anonymity. Users should be told the limits of privacy and anonymity to avoid giving the false impression that they can act with impunity.

Chapter 6

Tickets

An important building block for performing distributed identity transactions is a structured message, hereafter called a **ticket**. A ticket contains at least a random nonce, a logical timestamp, a local clock timestamp, an assertion, and a signature verification key. The ticket issuer appends a digital signature to the ticket. Tickets are a means of encoding information for communication between systems. The encoding and decoding of information in tickets is part of the presentation layer of the Internet.

The nonce provides a unique identifier for each ticket. The logical timestamp creates a partial ordering of tickets without assuming the existence of a globally synchronized clock. The digital signature ensures the integrity of the ticket and prevents the issuer from repudiating their responsibility for the ticket and the assertion it contains.

The logical timestamp is provided by a logical clock, which does not necessarily match the standard time clock. A Lamport timestamp is one way to implement a logical clock in a distributed system. Each node sets its clock to one more than the largest timestamp it has seen on an incoming message, and places its timestamp on each one of its outgoing messages. Lamport timestamps can be used to test whether one ticket arrived before another was issued, a requirement for causality. All tickets handled by a player can be ordered by their timestamp values, though tickets handled by different players may be concurrent (not

ordered relative to each other) [68].

The Lamport timestamp has at least two important purposes. First, the timestamp limits causality. If a player issues a ticket with a timestamp c_1, they cannot later claim to have relied on a ticket with a timestamp c_2, where $c_2 > c_1$. Second, the timestamp allows players to place temporal restrictions on transactions. For example, assume that a player A requests a platform B to close the account of A, and that B sends A a ticket confirming the closure of the account with the timestamp c_3. Any transaction in the account after c_3 would be inconsistent with the assurance of B that the account had been closed.

The local clock timestamp can be used to ensure freshness. For example, if a player issues a ticket as part of a transaction and the protocol requires the ticket to be returned to that player, then the player can check that the transaction has not timed out. Freshness is a common requirement in network security protocols [62].

For the sake of simplicity, the Lamport timestamp and the local timestamp collectively will be referred to as the **timestamp** hereafter. A timestamp thus includes both values, allowing it to be used in protocols that require either logical time or local time.

6.1 Related work

The tickets described in this chapter are inspired in part by the Kerberos distributed authentication system [90]. Kerberos has been widely deployed in local area networks. Because Kerberos is designed to work on insecure communication channels, it is possible to generalize Kerberos to work on the Internet. Chapter 9 develops this idea.

6.2 Purpose of Tickets

The purpose of a ticket is to serve as digital evidence that a player has made an assertion. A player can generate a ticket or receive a ticket. Beyond the immediate benefit of providing an assurance to the ticket recipient that an assertion is true, tickets can be archived in a

data store for later use to help resolve disputes. Not only is the recipient interested in being convinced that an assertion is true, the recipient also needs to retain evidence sufficient to convince a legal tribunal that the assertion is true. Tickets will be used to build the identity protocols described in later chapters.

As explained in Chapter 5, evidence is any data, document, or information that tends to make a condition more likely to be true. To ensure that tickets are widely accepted as strong evidence, digital signatures on tickets should be resistant to existential forgery. Resistance to existential forgery means that an attacker who tries to forge a ticket will find it prohibitively difficult to generate a ticket that appears to have been signed with a key that the attacker does not possess [51].

Digital signatures are commonly used in blockchain applications to cryptographically secure data records. Tickets are logically equivalent to blocks on a distributed private blockchain. In the same way that blockchains can be secured with digital signatures, tickets can also be secured.

6.3 Structure of Tickets

Let $TIC_X(a) = (h, a, SIG(S_X, < h, a >))$ where TIC_X is the ticket-generating function of player X. The **header** h of the ticket contains $< t, c, P_X >$, where t is the nonce of the transaction, c is the timestamp and P_X is the key used to verify the signature of the ticket. The assertion a is the **payload** of the ticket. The digital signature function $SIG()$ uses S_X, the protected secret of the player X, to sign the ticket. The digital signature function may also use a cryptographically secure hash function to ensure the integrity of the ticket header and payload.

The signature verification key P_X is an identity commitment for the player X. The signature of the ticket prevents the player X from repudiating the assertion in a ticket that he has issued.

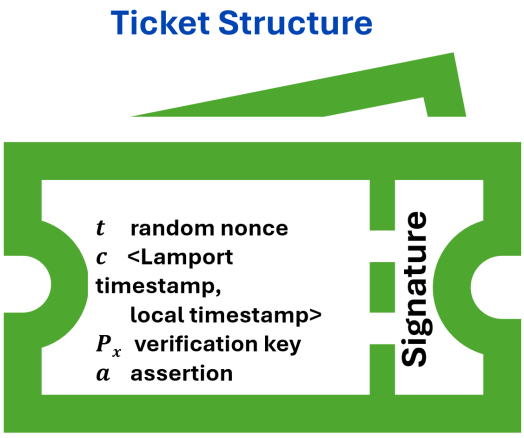

Figure 6.1: Ticket structure

6.4 Applying Digital Signatures to Tickets

The digital signature on a ticket must meet three requirements. As explained above, the signature must resist existential forgery. With high probability, a signature can only be generated by the player X who has the protected secret S_X that matches the corresponding identity commitment P_X. Second, the signature must be applied to the ticket in a way that protects the integrity of the nonce, timestamp, and assertion. Applying a cryptographically secure hash function to the data and then signing the hash is a standard method of preventing forgery [47]. Third, the signature must be verifiable by any player who receives the ticket using the identity commitment P_X in the header of the ticket. As a result, the signature provides strong evidence that the player associated with the identity commitment P_X made the assertion contained in the ticket.

6.5 Independent and Dependent Tickets

An **independent** ticket is a ticket that contains a proof without external reliance. For example, for the assertion $a =$"A, the issuer of this ticket uses P_A as an identity commitment," the ticket could take the form $T_0 = TIC_A(a)$ where A is the player and P_A is the ticket's

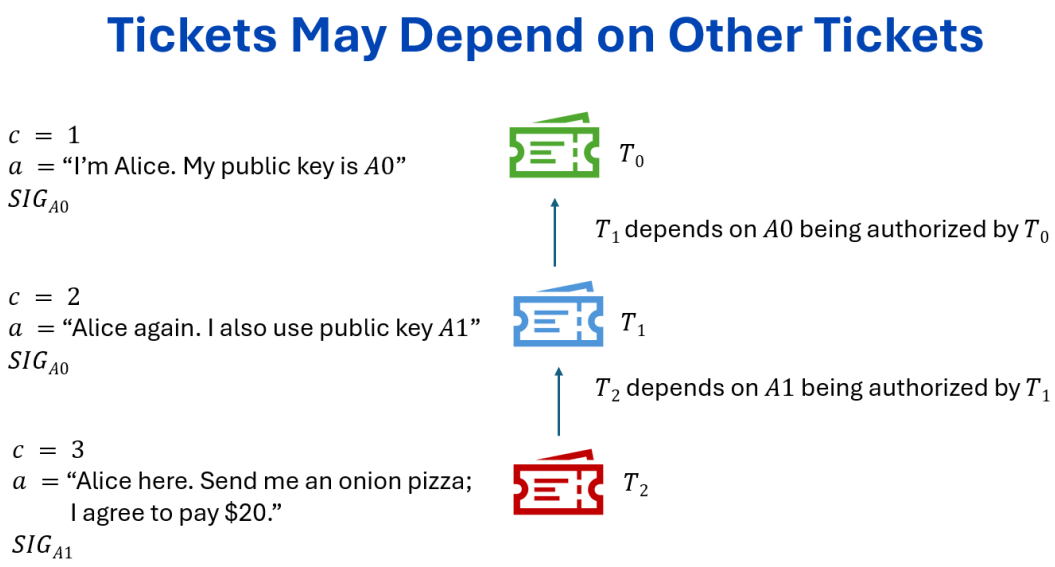

Figure 6.2: A green independent ticket, and blue and red dependent tickets.

verification key. A recipient B can verify that the P_A verification key in the ticket header is the same as the P_A key mentioned in the assertion. No reference to other tickets is necessary to determine whether the assertion m in T_0 is true.

A **dependent** ticket is one that requires reference to external tickets or data to prove the assertion it contains. Because Lamport timestamps create a partial ordering of messages (tickets) received and sent by a process (a player), a dependent ticket must have a timestamp larger than any ticket that the player claims to have relied upon in preparing the dependent ticket.

As an example of a dependent ticket, if a player A creates an account on a digital platform B, the player might register their verification keys P_{A0} and P_{A1}. The player A might later authorize a transaction, such as a purchase. Assume A has two devices. She should be able to use either one to generate a ticket T_A to authorize a transaction. One device might use P_{A0}, and the other might use P_{A1}.

Assume that a ticket T_A includes the verification key P_{A1}. The platform needs to do two things. First, the platform needs to verify the signature on the ticket using P_{A1}. Second,

the platform must determine whether P_{A1} is one of the verification keys that the player A has authorized. The platform will need to reference its data store to ensure that P_{A1} was authorized by the player. Performing a signature check to ensure the integrity of the ticket is only half of the job. The other half is to ensure that the signature verification key belongs to the player A, not an impostor.

Assume that P_{A1} has been properly authorized as an identity commitment for player A by ticket T_1 sent to a platform. The platform must save the ticket T_1 in case a ticket verified with P_{A1} is ever disputed. For example, if a ticket T_A includes the verification key P_{A1}, then ticket T_A depends on ticket T_1, because T_1 is necessary to prove the validity of T_A. T_1 proves that P_{A1} is an authorized identity commitment of A.

6.6 Tickets and Accountability

As explained in Section 4.3, accountability means that the system or protocol provides evidence that can be used to hold a wrongdoer liable for resulting harms. Tickets are a mechanism that provides accountability. Upon making an assertion, the player becomes responsible for the recipient's reliance upon the assertion. The ticket provides evidence of who made the assertion, what the assertion said, and when the assertion was made.

Why would a player ever want to take on the responsibility for making an assertion? When conducting business, buyers and sellers frequently make promises to each other. I might tell a contractor "If you repair my roof, I will pay you for time and materials." Profit is another possible motivation for making assertions. Signing a ticket is a service that can be sold for a fee, analogous to an insurance policy, a guarantee, or a notarization. For hundreds of years, guarantees have been recognized as a legitimate service available for a price. For example, when a bank lends money with real estate as collateral, they require the borrower to obtain title insurance to guarantee that the borrower has good title to the real estate used as collateral. The title insurance company asserts that the borrower has good

title, for a fee, and promises to pay the bank if the bank loses money because the assertion is false.

To avoid liability, a player B should not issue a ticket unless they can prove the assertion contained in the ticket. An element of such a proof could be a ticket issued to B by a second player T who B trusts. As an example, if Bob receives a ticket from Trent, whom Bob knows to be professional, of good reputation, and well capitalized, Bob can provide a ticket to Alice containing the assertion from Trent's ticket. Alice might know and trust Bob and be willing to rely upon the assertion he provides even though she does not know Trent and would not be willing to rely upon Trent's assertion without Bob's assertion.

In this way, a chain of trust can be formed from Trent to Bob to Alice, allowing a transaction to proceed. For example, Alice might be an e-commerce seller, Trent might be the purchaser, and Bob might be a bank. Trent sends Bob an assertion that he will pay for his purchase from Alice. Bob makes an assertion to Alice that Trent is good for the money. Alice can then proceed with the sale to Trent knowing that Bob will pay her losses if Trent does not pay.

If a loss occurs because of an assertion that is false, the chain of trust can be traced backwards from player to player based on the tickets that each has issued and received. If there is a break in the chain, the player who does not provide proof of their assertion can be held responsible. A player could be held responsible for making a reckless assertion without verifying the facts or for failing to save the evidence that supports their assertion.

Independent tickets are essential because these trust chains have to end somewhere. Independent tickets are a way to anchor a trust chain, as we will see in the federated identity design sketch in Chapter 7.

6.7 Tickets and Safety

As explained in Section 4.4, safety means that faithful players can defend themselves from claims using evidence. Tickets are a mechanism that provides safety. A player is safe when they can prove that their assertions are valid by presenting evidence in the form of tickets that they have saved. If a player C has relied on a ticket from another player A, then C can pass responsibility to A if a third party B accuses C of wrongdoing. For example, assume that Bob asserts that Alice approved a purchase from Trent and then Alice claimed that she never authorized the transaction. Trent might accuse Bob of making a false assertion. Bob would be safe if he could present Alice's ticket that approved the purchase. This evidence would defeat Trent's accusation against Bob. It would also put Alice on the hook for making the purchase.

If Bob relies on Alice when issuing a ticket and the transaction is challenged, Bob can use Alice's ticket to transfer responsibility for the assertion to Alice. The safety of Bob establishes the accountability of Alice. This passing of responsibility ends with an independent ticket that terminates the chain of trust, or with a break in the trust chain. When there is a break, the liability attaches to the party who cannot provide evidence to prove the assertion that they made. There is nothing new about the concept of passing responsibility.

6.8 Composition of Tickets

The composition of tickets occurs when an assertion includes a ticket (suitably encoded). For example, if $T_1 = TIC_A(m)$ and $T_2 = TIC_B(<$ "I endorse T_1", $T_1 >)$ where m is an assertion, then T_2 contains T_1. If Bob issues T_2, a protocol might define his assertion to mean that he has received T_1 and he asserts that the assertion m in T_1 is true. When a ticket T_2 contains another ticket T_1, the logical timestamp on the outer ticket T_2 must be greater than the logical timestamp on the inner ticket T_1. This prevents players from doing

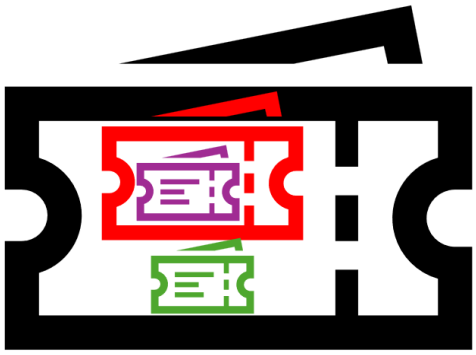

Figure 6.3: A ticket may contain other tickets.

something analogous to signing a blank check.

Composing tickets might not be desirable when a ticket contains sensitive information. When that is the case, it may be better for the ticket recipient to save the ticket, as a safety measure, and then issue a new ticket that contains just the necessary information, omitting any extra information. For example, Bob might receive from Alice a ticket containing a photo of Alice's driver's license. Bob might then issue a ticket asserting that Alice is older than 21. The recipient might only need to know that Alice is at least 21 years old and does not need to see Alice's picture or learn Alice's home address. In the event of a dispute, Bob could present the ticket he received from Alice to prove the assertion he made about Alice's age. To facilitate the submission of confidential evidence, courts have a procedure called "sealing" [80].

Chapter 7

Basic Identity

To build a digital identity infrastructure, we will need a data structure to represent the identity of a player. We also need protocols to build this identity. The data structure that follows is a private blockchain. The data have the logical structure of a blockchain because each ticket in the structure is digitally signed, and all but the first ticket refer to a prior ticket. The structure and associated protocol will be presented in its simplest form, without optimizations.

The following solution sketch simplifies real-world conditions to illustrate how identity can be proven within the Just Enough Trust framework. This solution is based on digitally signed tickets, though other implementations are possible. Any cryptographic scheme that satisfies the following condition could be used. The scheme must provide that each player has one or more public commitment values, each with a matching secret. A player who acts as verifier can use the public commitment value to construct puzzles that a player acting as prover can only solve with non-negligible probability if they possess the secret associated with the commitment value. When using public-private key pairs to perform digital signature operations, the commitment value is the verification key, and the secret is the private key. The puzzle could be a ticket containing a random nonce which the player must successfully sign to prove they know the secret.

7.1 First Authentication

Assume that Alice is a user and Carol is a digital platform. Alice has never done business with Carol before. She would like to open an account so that her data lodged with Carol could persist over multiple sessions.

Assume that Alice can randomly generate a unique verification key P_{A0} and a matching secret key S_A on the computing device she is currently using. Assume that Carol has access to a session identifier s that uniquely designates remote sessions for users currently connected to Carol's servers. As an example, s might be the socket number for a Transmission Control Protocol (TCP) connection [52]. This paragraph places some requirements on the environment that are necessary to run protocol 7.1.1 below.

Assume that Alice and Carol can use a communication channel that allows Alice to reliably identify Carol. In other words, we need not worry about Alice contacting an impostor posing as Carol. For example, Alice might connect to Carol via HTTPS [55], or Carol might have a fixed Internet Protocol (IP) address that Alice knows. The channel does not need to identify Alice to Carol, and the channel could be subject to eavesdroppers and man-in-the-middle attacks. Alice and Carol now execute the following protocol.

Protocol 7.1.1. *Basic authentication*

1. *Alice contacts Carol and requests to log in.*
2. *Carol sends Alice a ticket $T_1 = TIC_C(< s, m >)$, the network session identifier s, and $m =$"login", indicating that this is a login ticket.*
3. *Alice generates $T_2 = TIC_A(< T_1, P_{A0} >)$ and sends it to Carol.*
4. *Carol uses P_{A0} to verify the signature in T_2 and checks that P_{A0} is the verification key provided in the header of T_2.*
5. *Carol stores T_2 in Alice's account as proof that Alice authorized P_{A0} as her identity commitment, and that Alice knows the associated secret key S_A.*

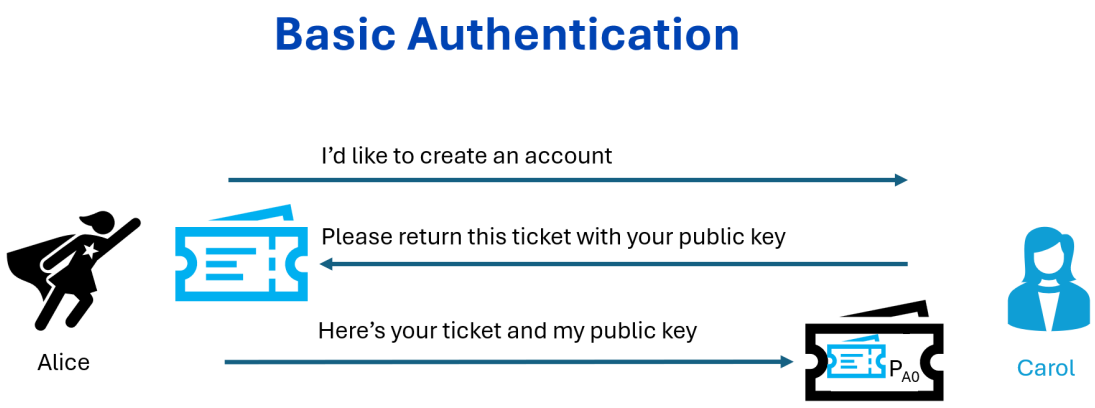

Figure 7.1: Basic Authentication

A protocol 7.1.1 transaction semantically means that Alice has told Carol that she is the person holding the secret key S_A that matches the verification key P_{A0}. No other player can obtain S_A due to the way protected secrets are defined. The protected secret S_A cannot be shared if the hardware security module on Alice's device works properly. This ticket T_2 is independent because Carol can use Alice's verification key in the ticket payload to verify Alice's signature. Carol does not need to reference any external data or tickets to verify the assertion in T_2. The ticket itself contains enough information to prove the assertion it contains. In the future, Carol can identify Alice by repeating the Protocol 7.1.1. Carol can thus associate persistent state data with her account. Alice's account is identified by P_{A0}. We can call P_{A0} Alice's **root key**. Carol can also acquire proof that Alice has authorized a transaction by running protocol 7.1.1 with m set to a string that includes the transaction details.

Protocol 7.1.1 satisfies Just Enough Trust because:

1. Choice: Alice has chosen to do business with Carol.

2. Accountability: For any transaction Carol claims to have been performed by Alice, Alice can demand that Carol present the signed ticket associated with that transaction.

3. Safety: If Carol has discharged her obligations to verify that transactions conducted

on Alice's account were authorized by Alice, Carol will be able to present a signed ticket for each such transaction. With all but negligible probability, no other player can forge a ticket from Alice because the secret key resides within Alice's device and has not been copied or transmitted.

4. Privacy: Carol has learned nothing about Alice other than the data that Alice knowingly provides to Carol, as long as Alice does not reuse P_{A0} for any transactions other than those with Carol. If Alice reuses P_{A0} on another platform, such as with Eve, there is a danger that Carol and Eve could gossip about Alice without her consent. When the use of P_{A0} is restricted to being used in a single place, P_{A0} is a hapax as defined in Chapter .

7.2 Adding Devices

Once Alice has an account with Carol, which is identified with Alice's root key P_{A0}, Alice might want to access Carol's platform with some of her other devices. She might also want to periodically rotate her keys, or she might dispose of her device and replace it with a new device. Assume that Carol will maintain a **keyring** for Alice, a list of Alice's authorized verification keys which serve as identity commitments for Alice.

Alice wants to add a new identity commitment to her account with Carol because Alice has begun using a new device. Alice generates a new verification key P_{An} and the matching secret key S_{An} on the new device. Alice and Carol run the following protocol.

Protocol 7.2.1. *Adding a New Key*

1. *Alice uses her new device to transmit P_{An} to one of her previously registered devices.*

2. *Alice uses that previously registered device to contact Carol and requests to add a new identity commitment via $T_0 = TIC_A(<"Addkey", P_{A0}>)$.*

3. *Carol sends Alice a ticket $T_1 = TIC_C(< s, m, P_{A0} >)$, with the session identifier s and m ="add key", indicating that this is a transaction to add a commitment to the account with root key P_{A0}.*

4. *Alice sends Carol the ticket $T_2 = TIC_A(< T_1, P_{An} >)$.*

5. *Carol verifies Alice's signature in T_2.*

6. *Carol stores the ticket T_2 in Alice's account as proof that Alice authorized P_{An}, and adds P_{An} to Alice's keyring.*

Protocol 7.2.1 satisfies Just Enough Trust because:

1. Choice: Alice has chosen to do business with Carol.

2. Accountability: For any key addition that Carol claims to have been performed by Alice, Alice can demand that Carol present the signed ticket associated with that transaction, which will include a P_{Aj} with $0 \leq j \leq n$. Alice can also demand to see a sequence of tickets showing that each verification key P_{Aj} was added through a signed transaction that verifies with a P_{Ai} with $0 \leq i < j$.

3. Safety: If Carol has discharged her obligations properly, she will be able to present a signed ticket for each such transaction adding P_{Ak} that verifies with P_{Aj}, $0 \leq j < k$. This chain will eventually lead all the way back to P_{A0} for which Carol holds an independent ticket. Nobody can forge any of Alice's n signatures because the secret keys reside within Alice's devices and have not been copied or transmitted.

4. Privacy: Carol has learned nothing about Alice other than the data that Alice knowingly provides to Carol, as long as Alice chooses each P_{Ak} such that it is a hapax, as explained above.

Once Alice has added additional keys P_{Ai} with $0 \leq i < n$ to her account with Carol, she can run Protocol 7.1.1 by substituting any of her P_{Ai} keys in place of her root key

P_{A0}. All of the keys work equally well, giving Alice the flexibility to use whichever of her devices she prefers.

7.3 Similarity to Blockchain

This system creates a chain of authorized keys, where the first key authorizes itself and then each subsequent key is authorized by a prior key. Each ticket that authorizes the addition of a new key is signed using a key contained in a prior ticket. Viewed this way, each ticket is a block and the signatures link the blocks together. For the sake of privacy, tickets need not be made public. In the event of a dispute, players can present the tickets as evidence to help prosecute or defend claims.

7.4 Invalidating Keys

Adding a feature to allow Alice to invalidate keys, such as when she discards one of her devices or rotates a key, makes Protocol 7.2.1 more complicated. First, it becomes necessary to utilize logical timestamps to ensure that transactions can be ordered. If a key could be invalidated, then the order of transactions matters. An authentication performed with an invalidated key is not valid. Hence, it becomes necessary to order the transactions performed by Carol and know which came first, the authorization transaction or the invalidation transaction.

Second, Carol must be able to return a signed ticket to Alice confirming that she has invalidated a key. With this collection of key-invalidation tickets in hand, Alice can hold Carol accountable if she subsequently authorizes a transaction with an invalidated key. Subsequent transactions by Carol can easily be recognized because they will have a higher logical timestamp.

Third, there is a possibility of wedged transactions where Alice might try to invalidate a key, but the transaction might not commit on Carol's end due to communication failure or

a faulty computing process. Consequently, Alice will need to have a way to meet face-to-face with Carol to clean up such a transaction. Implementing this capability is complicated in the real world and is best done with trusted agents. The next chapter will present the extensive infrastructure necessary to build out a full identity solution, including the trusted agents that would make this processing easier for the players.

Chapter 8

Trusted Agents

Trusted agents are extremely useful to facilitate distributed transactions. One feature of trusted agents is that they can provide local presence, facilitating face-to-face meetings. The local presence provides a bridge between the digital and physical worlds. A second feature is that they can adapt to local culture, customs, and laws, thus providing better service to their customers. A third feature of trusted agents is that they eliminate conflicts of interest.

The age-old question ”Who watches the watchers?” explains why each player should have their own trusted agent, rather than trying to agree on a single trusted agent. Systems that rely on a single authority, which includes most current federated identity systems, place too much trust in that single authority. Who watches them? With trusted agents, when each player can choose a trusted agent, there are at least two trusted agents who help to mediate each transaction. The watchers watch each other's players to ensure that nobody cheats [1].

[1] The concept of two watchers is suggested by the Watchbirds series of cartoons and childrens' books from the 1930's and 1940's, which remained culturally prominent for several decades. [11]

Figure 8.1: The benefits of trusted agents

8.1 Additional Services Like Account Sharing

In addition to the above benefits, trusted agents can provide additional identity services. For example, trusted agents could facilitate the sharing of accounts. When a married couple wants to jointly manage a credit card, but the credit card issuer only allows one login, that login could be controlled by the trusted agent, who then makes it available to either spouse. The technical possibility will be obvious when the UNS federated identity system is explained in Chapter 9, though an actual implementation has many legal considerations that are beyond the scope of this dissertation.

Some accounts are suited for joint custody, but others might be incompatible. For example, Netflix has tightened control over the sharing of accounts and charges extra fees for additional users [22]. A trusted agent offering joint custody as a service should be transparent and ensure that any account sharing is agreeable to all interested parties.

Beyond account sharing, there are situations in which a second party may have a legal obligation or the right to monitor the accounts of a first party. Parents should be able to supervise their child's online activity. A conservator might be appointed to manage the affairs of a vulnerable person. Law enforcement may have a court order that entitles them to search the data contained in an online account. An executor may have the right to manage

the accounts of a deceased person. The last two possibilities are specifically addressed in Chapter 14.

8.2 Local Presence

Throughout history, for hundreds or even thousands of years, trusted agents have been an important component of commerce and long-distance trade. For example, it is infeasible for a ship owner to meet his ship in every port to buy provisions and fuel. Consequently, local arrangements for cargo ships are handled by a network of ship agents hired by ship owners. Each port has ship agents to take care of the logistic details. Similarly, it is not feasible for a platform like Facebook.com to meet face-to-face with all customers to check credentials or to clear up wedged transactions. Amazon.com might have a better chance of providing local presence because it owns the Whole Foods chain of supermarkets.

Credential verification becomes important for verifying user attributes or certifying legal compliance. For example, if a user needs to prove that they are at least 21 years old to order a bottle of wine, the safest way to do that is to have a trusted agent check ID. Consider a bouncer who stands at the entrance to a busy club and checks IDs. The club owner trusts the bouncer to do the job properly, otherwise the club could be subject to expensive fines or get shut down. If we want to have a lawful Internet, we need the means to check that transactions and the parties thereto comply with the law.

Although online transactions can be documented with tickets, there may be times when a transaction goes outside the routine and has to be handled with a bespoke set of documents. In this case, a player can visit the physical office of a trusted agent and present legal documents, and the trusted agent can save those documents as evidence that they have met their obligations to the client.

8.3 The Fischer Problem

Wedged transactions are inevitable in a distributed system due to the Fischer problem. No matter how hard we try, any distributed identity system will suffer occasional consensus failures. When something bad happens, such as if the user is locked out of their account because the platform and the user no longer agree how to authorize access, there must be a way to resolve the disagreement. What if a user is robbed and simultaneously loses all of their authenticator devices? What if a user dies and their next of kin needs to enforce provisions in their will? To do that, the platform might need to inspect the will, a written document. There will always be situations where distributed systems cannot solve a problem online, but it would be redundant and impractical for each platform to maintain a network of local offices.

Trusted agents are a good way for platforms to establish synchronous communication with users to eliminate the Fischer problem, and to provide physical inspection of legal documents such as passports, wills, visas, and identity cards. In a face-to-face meeting, we assume that both parties instantly know if a message is received. Once a transaction is no longer distributed, the Fischer constraint no longer applies. At this stage, we do not need to specify what that protocol might look like, though a later chapter on key recovery will explore that problem.

If a face-to-face meeting cannot resolve the disagreement, the situation can escalate to become a dispute. A formal dispute resolution process, such as arbitration or a lawsuit, can be used to establish a result, even if the players do not cooperate. Agents are quite helpful in dispute resolution because such a process often requires physical presence at a hearing and a representative from each side who can knowledgeably testify about the dispute. A fact finding agent, such as an arbitrator, judge, or jury, can then decide the appropriate outcome and enforce it on the parties.

8.4 Professionalism

Another argument for trusted agents is that the agents can form an association and establish standard rules of conduct for how they will interact. As an example, lawyers act as agents for their clients, and they are subject to ethical rules of conduct established by their local bar association. Banks often act as financial agents for their clients and have numerous associations with rules and standards, not to mention the regulation that ensures an orderly marketplace.

There are many historical precedents for the participation of an agent acting on behalf of each party. In legal matters, it is common for each party to have their own lawyer. It would often be a conflict of interest for a lawyer to act on behalf of two parties who have opposing interests [53]. In a real estate sales transaction, the buyer and the seller may each choose a different notary to verify their signature on the deed that conveys title to the property. In international trade, the buyer and the seller often utilize different banks. The buyer's bank may issue a guarantee of payment in the form of a letter of credit, and the seller's bank will confirm the letter of credit, giving the seller an assurance that the buyer's bank will make good on its promise to pay if the goods are shipped according to contract terms. This banking protocol was first standardized by the International Chamber of Commerce in 1933 [92].

The notarial function, the creation of reliable written records, has existed for at least two thousand years [39]. It is remarkable that the Internet lacks an analogous standard for establishing reliable transaction records, although blockchains are a first step towards such a system.

Lastly, the implementation of cryptographic protocols is tricky. Coding a cryptographic solution well requires specialized expertise that many platforms and virtually all users lack. Trusted agents can make the necessary investments to build a secure, scalable, and stable identity infrastructure. Because trusted agents can serve numerous clients, their agency is a means of scaling expertise and making a profit.

Chapter 9

Federated Identity

Federated identity is an identity infrastructure system that provides access to multiple systems through a central authentication service. Federated identity systems are sometimes called single-sign-on (SSO) systems. For example, Yale's Central Authentication Service allows a user to log in with a single account to gain access to any of a multitude of systems run by Yale University. Google offers a federated identity system. A user can link their Google identity to a third-party platform and gain access by logging into Google. Facebook and Apple offer similar systems. See Figure 1.1 for an example showing each of these three providers.

An important feature of federated identity systems is that the identity receiver, typically a platform, does not need to keep track of each player's identity commitments. A central authentication server keeps the identity commitments, issues the puzzles, checks the responses, and then provides the identity receiver with a consistent identifier for the player.

An obvious problem with federated identity systems is that a consistent identifier can be used by two or more identity receivers to exchange information about the player without the player's consent. Moreover, a central authentication service might have interests counter to the platform or players and might use the knowledge gained about the players' log-in

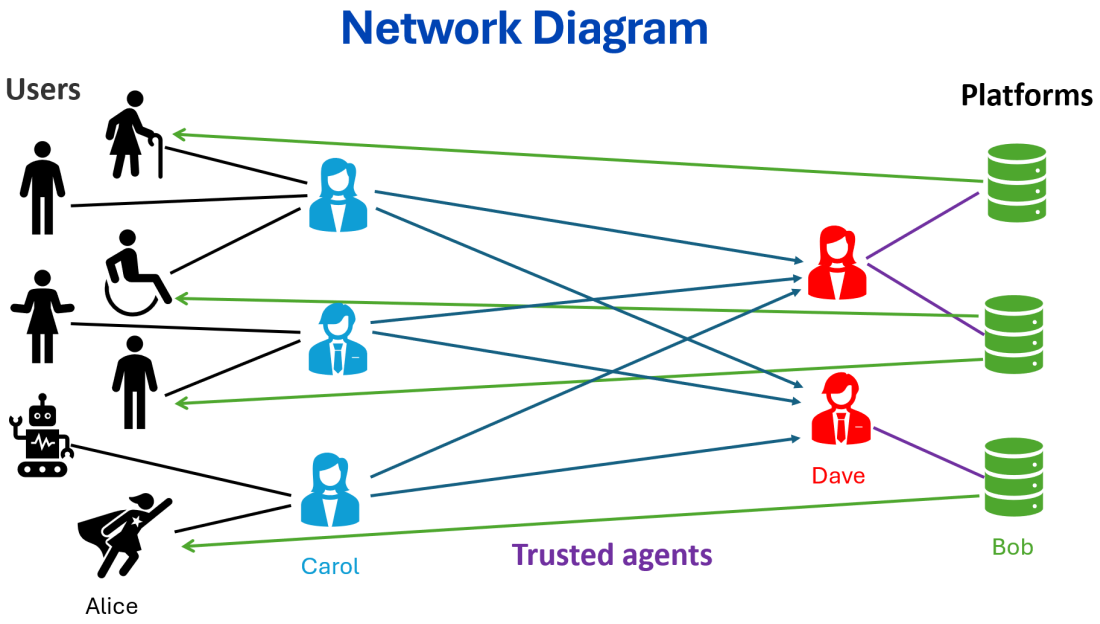

Figure 9.1: Network of trusted agents as a foundation for trustworthy transactions

behavior for adversarial purposes. Consider a hypothetical federated identity service called the Boogle login, where Boogle is also a major advertising exchange. When a user logs in with Boogle, the transaction produces information that Boogle can add to the user's profile and use to target the user with ads. A bakery should hesitate to use log-in with Boogle as its federated identity solution since Boogle can exploit its knowledge of who is logging in to the bakery's website to target the bakery's customers with ads and offers from a competing bakery.

9.1 The MN Problem

The MN problem refers to the number of ways of pairing an object from a set of size M with an object from a set of size N. It was addressed in 1961 by UNCOL, the Universal Computer Oriented Language. UNCOL aspired to solve "a fundamental problem of the digital data processing business: automated translation of programs from expressions in an ever increasing set of problem oriented languages into the machine languages of an expand-

ing variety of digital processing devices." UNCOL consisted of a compiler front end that converted a computer language into UNCOL, and a back end that would convert UNCOL into machine code. This solved the problem of having compilers for M languages running on N different computer architectures. Instead of having to generate $M * N$ compilers, UNCOL required M front ends and N back ends, for a total of $M + N$ programs [89].

9.2 Federated Identity and the MN Problem

Federated identity is motivated by the need to simplify the management of user authentication. If a user has M accounts on different services and N computing devices, they would potentially need to generate $M * N$ secret keys and matching verification keys to maintain security and privacy. Similarly, if services use passwords, then the user must generate and maintain M unique passwords and synchronize them with N devices, under the assumption that no one can actually remember so many unique high-entropy passwords.

With federated identity, it becomes possible to solve the "MN" problem. Instead of managing $M*N$ account-device pairs, the user can generate N secrets, one for each device, and use them to authenticate to the federated identity service. The service then provides each of the M platforms with its unique identifier for the user. In this way, there are $M+N$ relationships to manage, which is substantially less complex than the $M*N$ account-device pairs.

Consider what happens if a user registers N keys with each of M services and then loses a device. They must visit all the M services to cancel the verification keys that were on the lost device. This would be impractical. It is much easier to visit the federated identity server, cancel the verification key there, and then relax knowing that the canceled key could no longer be used to access any of the services.

On the platform side, federated identity can reduce the burden of building and maintaining a robust user authentication system. Instead of each platform needing to build com-

plicated software infrastructure services, most of the work can be offloaded to a federated identity provider.

9.3 Difficulties with Existing Federated Identity Services

One challenge with existing federated identity systems is that not all users have the same federated identity accounts as those accepted by the platforms they use. For instance, a user might not like to use Login with Google because they do not trust Google, a major online advertising company that is known to traffic in user information and exploit it for financial gain. If a platform chooses Google login, then the user and the platform are in a standoff. If a platform has to support multiple federated identity services, this defeats the purpose of using federated identity to simplify identity management, and it still falls short of a complete generalization. In a complete generalization of the federated identity problem, the platform can accept any identity provider the user chooses.

Another problem with federated identity systems is the lack of isolation, or viewed another way, too much centralization of power. For example, the US federal government uses ID.me, a technology start-up company, as its federated identity provider [61]. But what if there is a failure with ID.me? A failure could be technical, or there could be a failure in the governance of ID.me. What if ID.me decides, for whatever reason, that a citizen is an "unperson"[1] and cuts them off? That citizen can no longer access their online accounts with any federal agency. What if ID.me is purchased by an egomaniac billionaire who compromises its operations for political ends?

One possible solution would be to allow the user to choose a trusted agent to act as their identity provider, and for the platform to independently choose a trusted agent to act as identity recipient. In this way, the two trusted agents could work together to authenticate the user. Importantly, each user then has freedom of choice, and so does the platform, to

[1] "Unperson" was coined in George Orwell's book "1984" and means individual who has been erased from society.

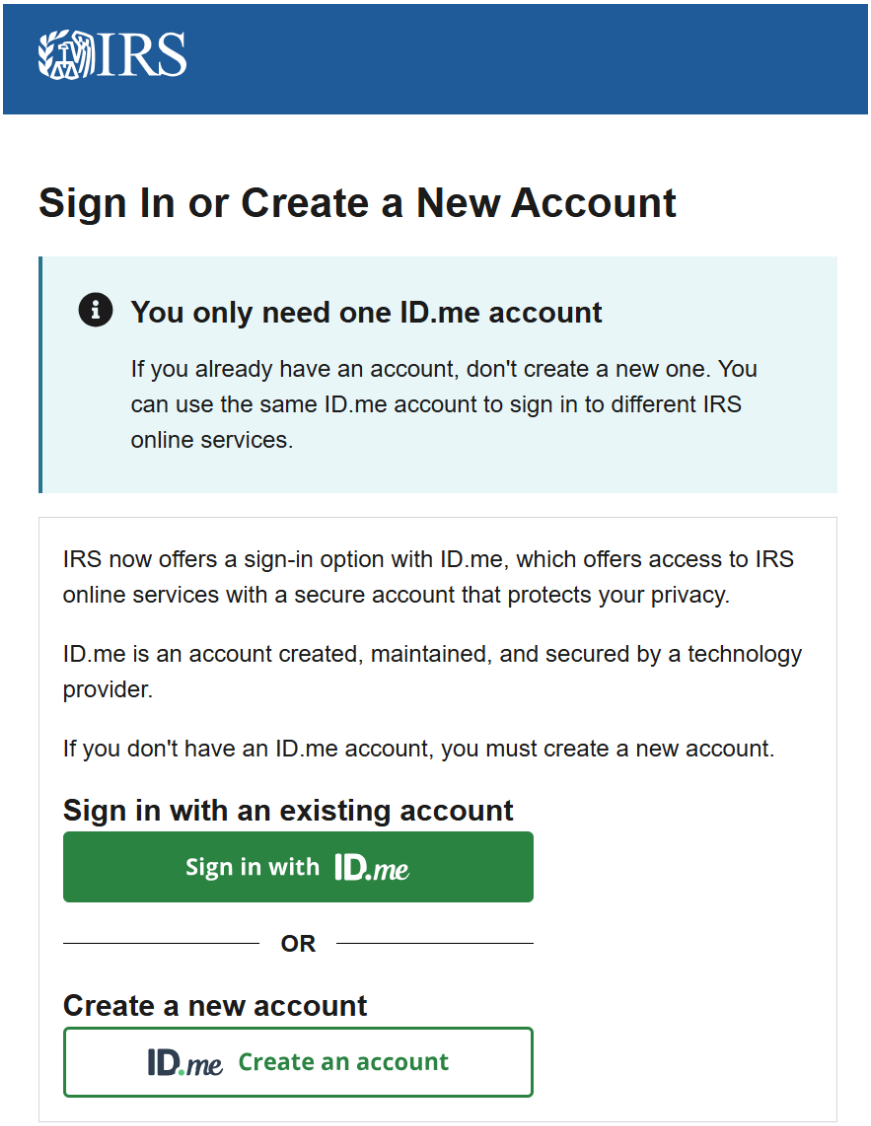

Figure 9.2: Example of a government login screen using ID.me [8]

pick the trusted agent they like. In effect, the central authentication service is split into two pieces, one chosen by the user and the other chosen by the platform.

In the next section, an example of such a system is presented. Its network topology consists of a network of trusted agents.

9.4 Design Sketch for UNS

The Universal Name System (UNS) is an experimental identity infrastructure service that aims to provide a uniform login experience on the Internet, consistent with the principles of Just Enough Trust. US Patent 11,570,163 provides detailed UNS specifications [65]. UNS is implemented through a network of trusted agents. This design sketch provides a high-level view of the main protocols needed to implement such a system. The actual implementation has many details that have been elided for the sake of clarity.

Assume that each player chooses a trusted agent they prefer. The trusted agents promise to preserve the user's privacy. More than one player can work with each trusted agent. Assume also that the trusted agents are licensed by a professional organization, are regulated by their governments, are members of a global association of trusted agents, and that the association has a set of rules for how trusted agents must behave. The rules include a requirement that each trusted agent provides a financial security to ensure it's good faith and to compensate any party injured by a trusted agent's breach of faith. When there is a dispute about who to blame for an authentication error, the rules also specify a dispute resolution process that is mandatory for trusted agents. The dispute resolution process includes one or more face-to-face meetings that allow the disputants to exchange information nearly instantly and to know without a doubt whether their messages are received or not. In other words, the dispute resolution process can solve the Fischer problem.

In a federated identity user authentication transaction, there are at least four players:

1. Alice, a player who wants to prove her identity.

2. Bob, a player who wants to receive proof of Alice's identity.

3. Carol, a trusted agent chosen by Alice.

4. Dave, a trusted agent chosen by Bob.

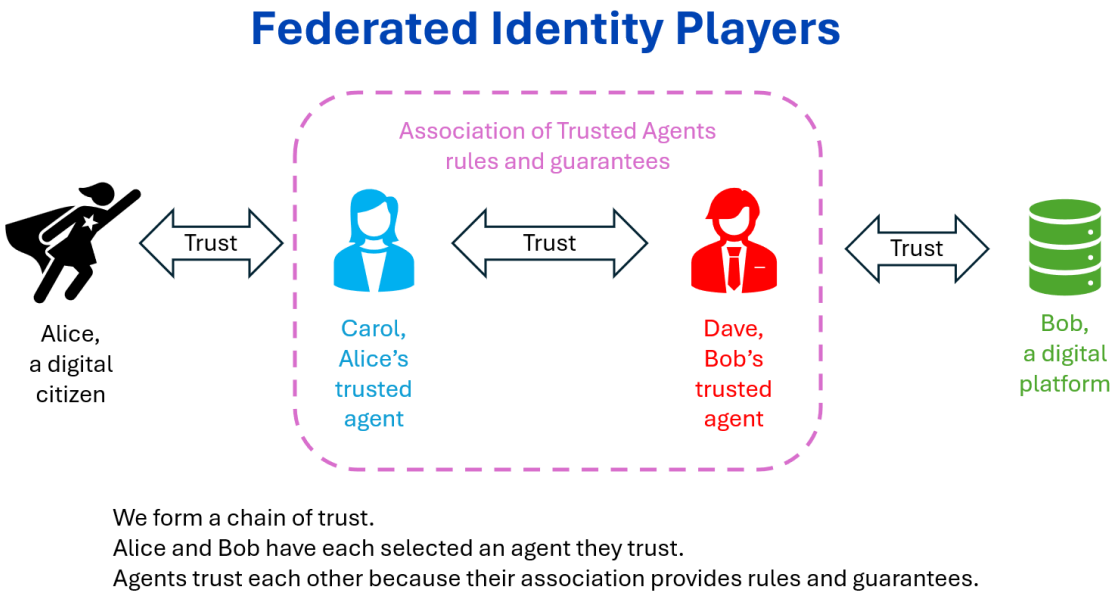

Figure 9.3: Federated Identity players

Because Carol and Dave are members of the trusted agents' association, they are on a distribution list maintained by the association's executive committee, which periodically distributes a routing list $\mathbf{M}$ that includes the names, verification keys, and network addresses of all trusted agents. If a routing list is not received on schedule, assume that Carol and Dave have the ability to visit the executive committee office in person to pick up the latest routing list.

Assume that Alice and Carol agree on an authentication protocol $\mathcal{Q}$ that Alice will use to prove her identity to Carol. The protocol described in Chapter 7 is one such example. If Alice and Carol choose a different protocol, it must at least be able to carry a data payload containing an admission ticket and a network connection identifier.

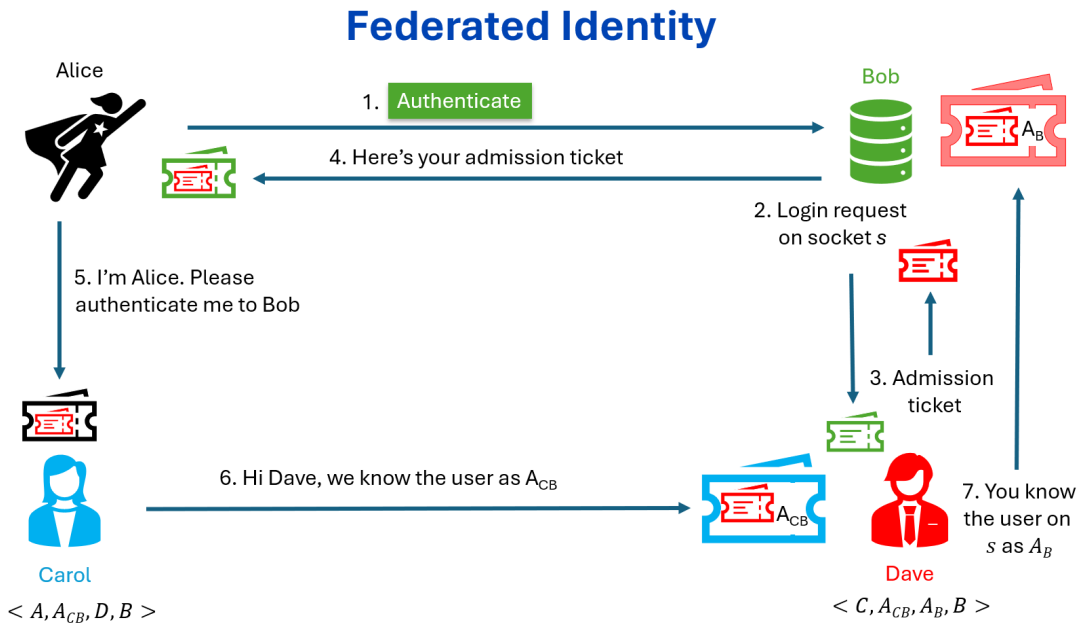

Figure 9.4: Federated authentication

In addition, assume that each player saves the tickets that they receive in order to provide an audit trail. The audit trail provides evidence to help resolve any disputes related to the authentication transaction.

9.4.1 First Authentication

The first time Alice attempts to authenticate with Bob, the system must provision a hapax identifier for Alice that will be used by Bob only. The protocol is as follows:

Protocol 9.4.1. *Binding a name*

1. *Alice requests to log in with Bob.*
2. *Bob requests and obtains an admission ticket* $T_1 = TIC_D(< s, B, \text{“Admission”} >)$ *from Dave containing the Alice-Bob session identifier* s *(as seen by Bob) and Bob's identifier* B*.*
3. *Bob sends* $T_2 = TIC_B(T_1)$ *to Alice.*

4. *Alice uses protocol $\mathcal{Q}$ to prove her identity to Carol with T_1 included in the payload.*

5. *Because Carol has not previously helped Alice authenticate with Bob, Carol now generates a new hapax A_{CB} for Alice that will only be shared with Bob's trusted agent D (Dave). Carol saves $< A, A_{CB}, D, B >$ in her data store for future reference, where A is the root identifier of Alice.*

6. *Carol generates a new ticket $T_C = (< A_{CB}, C, T_1 >)$ and sends it to Dave.*

7. *Dave uses the routing list $\mathbf{M}$ to look up Carol's verification key P_C and verifies the ticket signature.*

8. *Dave sees that A_{CB} is a new hapax. He generates another hapax A_B to identify Alice with Bob.*

9. *Dave records $< C, A_{CB}, A_B, B >$ in his data store, recording that Carol is the custodian of A_{CB}.*

10. *Dave sends $T_D = TIC_D(< T_1, A_B >)$ to Bob.*

11. *Bob verifies Dave's signature on T_D and associates Alice's identifier A_B with the remote session s contained in T_1.*

During this protocol, Bob does not learn anything about Alice other than a hapax, a name that is used nowhere else. Therefore, protocol 9.4.1 preserves privacy. In addition, the hapax identifiers A_{CB} and A_B provide isolation. Carol does not learn A_B, and Bob does not learn Alice's verification keys. A single breach of Carol's or Bob's system does not provide enough information for an attacker to link Alice to her account with Bob.

9.4.2 Subsequent Authentications

Upon a subsequent attempt by Alice to authenticate with Bob, the protocol must provide Bob with the same hapax identifier A_B as the first contact between Alice and Bob.

Protocol 9.4.2. *Looking up a name*

1. *Alice requests to log in with Bob.*
2. *Bob requests and obtains an admission ticket* $T_1 = TIC_D(< s, B, \text{“Admission”} >)$ *from Dave containing the Alice-Bob session identifier* s *(as seen by Bob) and Bob's identifier* B*.*
3. *Bob sends* $T_2 = TIC_B(T_1)$ *to Alice.*
4. *Alice uses protocol* $\mathcal{Q}$ *to prove her identity to Carol with* T_1 *as the payload.*
5. *Because Carol has previously helped Alice authenticate to Bob, Carol finds* $< A, A_{CB}, D, B >$ *in her data store.*
6. *Carol generates a new ticket* $T_C = TIC_C(< A_{CB}, C, T_1 >)$ *and sends it to Dave.*
7. *Dave uses the routing list* $\mathbf{M}$ *to look up Carol's verification key* P_C *and verifies the* T_C *ticket signature.*
8. *Dave finds* $< C, A_{CB}, A_B, B >$ *in his data store and verifies that Carol is the custodian of* A_{CB}*.*
9. *Dave sends the ticket* $T_D = TIC_D(< T_1, A_B >)$ *to Bob.*
10. *Bob verifies Dave's signature on* T_D *and associates Alice's identifier* A_B *with the remote session* s *contained in* T_1*.*

The protocol is the same as the protocol 9.4.1 except that instead of generating a new hapax, Alice's trusted agent Carol sees that Alice has authenticated with Bob previously because Carol has a saved tuple $< A, A_{CB}, B >$. Alice reuses A_{CB} as a hapax identifier for Alice by sending T_C to Dave, who then checks that Carol is the custodian of A_{CB}, and then provides A_B to Bob.

This protocol ensures that Bob will always receive a consistent hapax identifier A_B for Alice, allowing him to maintain a consistent account for Alice. As with protocol 9.4.1, this protocol preserves privacy because Bob learns nothing about Alice other than A_B, an identifier that is not used anywhere else.

9.5 Logging in to the Internet

After a player A authenticates with their trusted agent C, that authentication should remain valid for the duration of the player's on-line session, subject to a reasonable time limit. Modern computing devices such as desktop computers and mobile phones automatically lock if A stops interacting with the device. Depending on the circumstances and the preferences of A, the locking of her device could terminate the authenticated session with C. Existing identification technologies that respect privacy, such as session cookies, are widely used to implement this type of behavior [33]. Thus, after A has authenticated with C, she can visit additional sites or connect to additional online services without having to re-authenticate with C. Logging in to those sites could be done with a single click, authorizing her trusted agent to provide her site-specific hapax to the site being visited.

How would this be implemented? Upon a player A authenticating with her trusted agent C via a browser, A's browser would receive and store a session cookie containing a ticket $TIC_C(< s, A >)$ that identifies A to C and also contains s the network session identifier for the connection between A and C. The browser could maintain that session live as long as the player remained active on the device, after which time the browser would terminate the session and the cookie would be discarded.

Whenever player A navigates to a login page on a site B, their browser would automatically receive T_1, an admission ticket for B as described in Protocol 9.4.1 or 9.4.2. To login, A clicks a button in the browser that sends T_1 to her trusted agent C. The trusted agent C then checks the session cookie saved by the browser of A. After verifying that

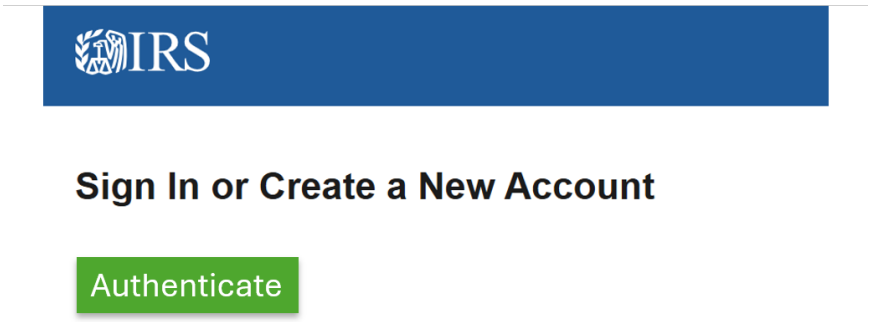

Figure 9.5: What a log in screen could look like with UNS

the cookie remains valid, C continues with the rest of the authentication protocol without requiring any further effort from A. This exemplary implementation would require a relatively minimal extension of the browser.

9.6 Accountability and Safety

Because the basic authentication protocol 7.1.1 provides accountability, the federated protocols 9.4.1 and 9.4.2 also provide accountability. First, Carol only provides A_B to Dave after receiving the proof in the form of a ticket from Alice. Carol can be asked to produce that ticket later, and if she cannot, then she can be held accountable. Second, Dave only provides A_B to Bob after receiving the proof in the form of ticket T_C from Carol. Dave can be asked to produce that ticket later, and if he cannot, then he can be held liable. Third, Bob can be asked to produce the ticket T_D, and if he cannot, then he can be held liable.

Even if Alice's hapax leaks to Eve, and even if Eve is a malicious trusted agent, she cannot produce a false authentication with A_B because Dave will only accept tickets containing A_B from Carol. A supplemental protocol could provide Carol the ability to release A_B and transfer it to another trusted agent, such as Felix. Essentially, Alice must ask Carol to release one or more of her identifiers to Felix. Alice saves the ticket and notifies each trusted agent to whom Alice has previously sent authorizations to take note that Felix will be processing transactions on behalf of Alice thereafter.

Because tickets are used at each stage of authentication, for the same reasons that protocols 9.4.1 and 9.4.2 provide accountability, they also provide safety. Whether Bob, Dave, or Carol is accused of breaching trust in a specific transaction, they can successfully defend themselves by producing the relevant ticket.

Protocol 9.6.1. *Transfer of hapax custody*

1. *Alice authenticates with Felix and asks him to serve as her trusted agent and to receive one or more hapax identifiers from Carol. Alice provides Felix with at least one of her identity commitments* P_{Ax}.

2. *Alice authenticates with Carol and sends ticket*
   $T_A = TIC_A(<$ *"Transfer my identity commitments to Felix"*$, \{ P_{Ai} \} >)$ *that requests transfer of one or more of her hapax identifiers* P_{Ai}. *Alice signs* T_A *so that the signature can be verified with* P_{Ax}.

3. *Carol sends a signed transfer ticket* $TT = (< T_A, \{ < h_i, W_i, Y_i > \} >)$ *containing the set of hapax identifiers that Alice wishes to transfer to Felix. Each* h_i *is a hapax used to identify Alice with a platform* Y_i *and its trusted agent* W_i.

4. *Felix verifies Alice's signature on* R *with* P_{Ax} *and records the tuples* $< A, h_i, W_i, Y_i >$ *for future use.*

5. *Felix sends Carol a signed ticket* $TT' = TIC_F(< TT, F >)$ *to confirm that he is now responsible for the hapax identifiers specified* $\{ H_i \}$.

6. *Carol saves* TT' *as a proof that she is no longer responsible for* $\{ H_i \}$.

When hapax identifiers are transferred from Carol to Felix, a clear record is created in the form of a signed ticket TT and a signed receipt TT'. Based upon these records, Carol and Felix can be held responsible for any hapax provided during the time period when they held custody. Here again, the logical clock proves useful because it allows tickets to be

ordered relative to one another. Likewise, Carol and Felix are safe from false accusations because they receive tickets that they can use to show when they were the custodian of the hapax in question.

9.7 Managing the Fischer Problem

Federated identity protocols are highly dependent on the reliable delivery and processing of messages between players. In any distributed implementation, transactions will occasionally enter a wedge state, where the participating players do not have a consistent view of the status of the transaction. This is a widespread problem with existing single-sign-on systems [71]. When this happens, players may need to meet face-to-face to eliminate the distributed property. Once they have synchronous communications they can re-establish consensus either by negotiation, or by arbitration before a neutral fact finder.

Chapter 10

Linked Identity

The federated identity system described in Chapter 9 can provide proof that a user is the same one who initially registered an account, not that the user was a specific legally recognized person. For some use cases, it is necessary to tie a digital identity to a legally recognized person or to certify the attributes of a player as they appear in the player's legal documents. Some use cases may require the ability to **unmask** a pseudonymous player in response to a court order by providing the court with a copy of the player's legal identity documents. A **linked identity** is a digital identity in which proof can be provided to tie the digital identity to a legal identity.

One benefit of linked identities is that they provide backward compatibility between a digital identity system and legacy forms of identification based on physical documents. Because it could take many years for society to shift to digital identification,[1] linked identities can provide an important bridge between digital and physical identification.

To generate a linked identity, the player may need to visit the physical premises of their trusted agent and present a legal identification document, such as a passport. During the visit, the trusted agent can compare a photo in the legal document with the observed player and determine whether the document belongs to the player. Banks would be ideally positioned to act as trusted agents for this purpose because they already perform legal identity

[1] Notably, Estonia established a national digital identity over 20 years ago [4].

verification and have networks of secure physical locations with networked computer terminals to collect data. The trusted agent would store a copy of the identity document, such as a scan or photograph, to ensure its own safety if the linked identity were ever challenged or subpoenaed by a court having jurisdiction over the trusted agent.

A coincidental benefit of linked identities based upon physical meetings between players and trusted agents is that the trusted agent can observe whether the player is subject to improper compulsion, the victim of a scam, or preparing to commit a crime. Society has widely recognized the potential benefits of enlisting front-line workers in bolstering security. For example, banks in the United States are required to verify customer identity and monitor lists of suspected terrorists [74]. Tourism and hospitality personnel could be trained to recognize and report human trafficking, which often becomes involved in these types of business [76]. Notaries rely on physical presence to detect and deter fraud [54]. Such safeguards would be difficult to implement through digital means alone. By combining a physical presence, greater resistance is established against threats of compulsion, improper influence, fraud, and criminality.

Likewise, a physical meeting serves as a deterrent to attempted document forgery. Currently, on-line platforms often try to verify legal attributes by requesting a scan of users' legal documents. This method admits the possibility of forgery through the use of photo editing software. When a user must enter a physical office and present a physical document, they are in direct peril of arrest if they attempt to pass forged documents.

Linked identities are the first step in establishing a system to detect and resist Sybil attacks. A protocol for Sybil account detection is presented in Chapter 12 based on linked identities.

10.1 Pseudonym generation

A linked identity consists of a pseudonym that has a mathematical relationship with a legal identifier. For example, a linked identity pseudonym could be generated by blinding a hash of the user's passport number concatenated with the name of the issuing country, $BL_x(H(N.C))$. To prevent an adversary from deriving the legal identity from the pseudonym, a blinding function BL_x is used where a key x is kept secret. An adversary who does not know the secret x cannot calculate the blinded value of a legal identifier. Chapter 11 presents the mathematical description of a blinding function.

A pseudonym used with more than one platform could be used as a join key that would allow the platforms to exchange information about the player without consent. One way to ensure that a linked identity protocol preserves privacy is to choose hapax pseudonyms. Rather than choosing a random hapax as explained in Chapter 9, each hapax could be generated by blinding a legal identifier with a different randomly selected x for each platform that will receive a pseudonym. Because no two users can have the same legal identifier, the risk of a pseudonym collision between users is negligible if the blinding function has a negligible risk of collisions. For convenience, such a hapax pseudonym can be called a **linked hapax** or a **linked identifier**.

Assume that there are four players.

1. Alice, a user who wants to log in to Bob's platform.

2. Bob, a platform operator.

3. Carol, who serves as Alice's trusted agent.

4. Dave, who serves as Bob's trusted agent.

The protocols proceed along the lines of those detailed in Chapter 9, except that the method of generating the hapax takes as input a legal identifier. Bob can require users to provide linked identities. Linked identities might be mandatory for all users or might be

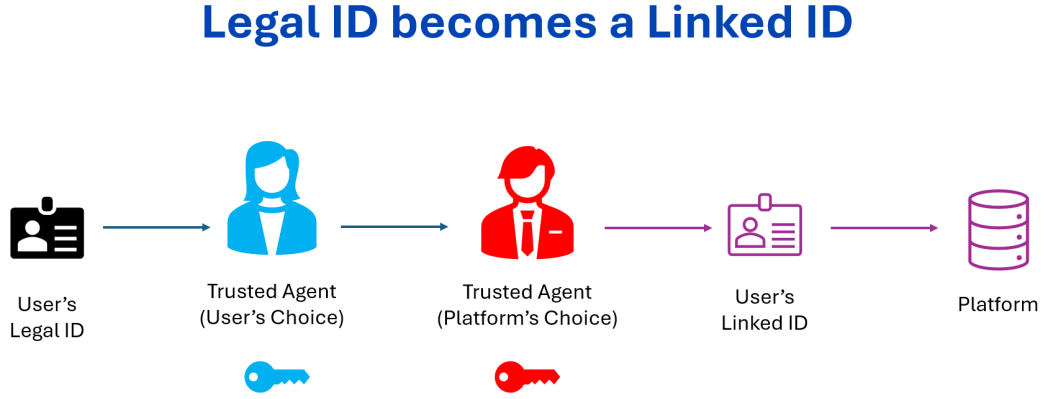

Figure 10.1: Linked identifier generation

an optional way to gain increased privileges. Bob will be informed when a user provides a linked identity.

Protocol 10.1.1. *Generating a Linked Identifier*

1. *Alice meets with Carol and Carol inspects Alice's documents, recording* L_A*, a value that is a function of Alice's legal identification document.*

2. *Alice performs a digital authentication for Carol as explained in Chapter 7. Carol keeps the tickets Alice provides as proof of Alice's identity commitments.*

3. *Alice requests to log in with Bob for the first time, as explained in Chapter 9 and receives* T_1 *from Bob. (Bob gets* T_1 *from Dave, a ticket containing a unique nonce and Dave's identifier.)*

4. *Alice sends* $T_2 = TIC_A(< T_1, "linkedID", B >)$ *to Carol, indicating that she wants to use a linked ID to authenticate with Bob.*

5. *Carol checks to see whether she has already created a secret key* u_B *to blind identifiers intended for Bob. If Carol does not already have* u_B*, she generates* u_b *at random*

and saves it.

6. *Carol calculates* $A_{Bu} = BL_{u_B}(L_A)$.

7. *Carol sends* $T_3 = TIC_C(< A_{Bu}, \text{“linked ID”}, T_1 >)$ *to Dave.*

8. *Dave checks to see whether he has already provisioned a secret key* v_B *to blind the linked identifiers for Bob. If Dave does not already have* v_B*, he generates* v_b *at random and saves it.*

9. *Dave calculates* $A_{Buv} = BL_{v_B}(A_{Bu})$.

10. *Dave sends* $T_4 = TIC_C(< A_{Buv}, \text{“linked ID”}, T_1 >)$ *to Bob.*

11. *Dave records that* C *is the trusted agent of the player known as* A_{Buv}.

12. *Bob now recognizes Alice with the persistent identifier* A_{Buv} *and knows that this is a linked identity.*

Because both Carol and Dave conduct blinding, the property of isolation limits the damage from a single breach. If one of u or v is exposed, that is not enough information to un-blind A_{Buv}, the linked identifier of Alice. Moreover, Carol does not learn A_{Buv} and Dave does not learn L_A. A player cannot leak information that they never acquire.

10.2 Certificates of Compliance

A certificate of compliance is an assertion that a trusted agent has inspected a digital or physical identity document of a player and that, based on that information, the player is in compliance with some law for some purpose. For example, a simple certification of compliance could assert that a user is at least 21 years old, and is thus allowed to view on-line content related to alcoholic beverages and place orders for delivery in the United States.

A more complicated certificate of compliance could be an assertion that a player buying cryptocurrency on an exchange is not on the Office of Foreign Asset Control (OFAC) sanctions list [7]. In addition, or in the alternative, a trusted agent could certify that a user is not a resident of a sanctioned country.

Assume the same four players as above and assume that Alice has already authenticated with Bob using her linked identity, which she has established with Carol. Carol thus has a copy of Alice's legal identity documents.

Protocol 10.2.1. *Certificate of compliance issuance*

1. *Alice wishes to make a transaction with Bob.*
2. *To comply with laws or regulations, Bob needs to verify a legal fact such as Alice's citizenship.*
3. *Bob obtains a ticket T_1, which contains a unique nonce, from his trusted agent Dave.*
4. *Bob issues a ticket $T_2 = TIC(< r, T_1 >)$, where r is a request for a proof of citizenship, and sends it to Alice.*
5. *Alice issues ticket $T_3 = TIC(T_2)$, confirming her consent to provide proof of her citizenship to Bob, and sends it to Carol.*
6. *Carol verifies Alice's signature and unpacks the tickets composed in T_3 to learn what proof is requested and that she should send this proof to Dave for Bob.*
7. *Carol issues ticket $T_4 = TIC(< A_{Buv}, \pi >)$, where π asserts Alice's citizenship based on the legal documents Alice has presented to Carol, and A_{Buv} is the hapax used to identify Alice to Bob, and sends T_4 to Dave.*
8. *Dave verifies Carol's signature on T_4 using the routing list* $\mathbf{M}$*, and checks that Carol is the trusted agent for the player identified as A_{Buv}.*

9. *Dave creates a ticket $T_5 = TIC(< A_{Buv}, \pi >)$ and sends it to Bob.*

10. *Bob verifies Dave's signature on T_5 and records the proof π of Alice's citizenship.*

Each ticket recipient saves the tickets received to ensure safety from future challenges.

Protocol 10.2.1 preserves privacy because Bob learns nothing more than the fact that Alice consented to share π. If Bob has a good reason to know Alice's citizenship, he can get a reliable assurance from his trusted agent Dave without learning Alice's passport number, seeing her photograph, or discovering her date of birth. By limiting the information disclosed to Bob to just what is needed, Alice's privacy is much better protected than if she were to send Bob a picture of her passport. In addition, Bob is better protected because he has a ticket containing proof and he can hold Dave responsible in case of a future challenge.

10.3 Unmasking as a Form of Accountability

Accountability flows not only from platforms to users, but also from users to platforms. In the above example, if Bob is subject to an investigation and a court demands that Bob reveal the identity of Alice, Bob can provide a ticket referencing Dave as a trusted agent who knows how to find Alice, the person who goes by the pseudonym A_B on Bob's platform. Dave can refer to Carol as a trusted agent who has certified Alice's citizenship and knows Alice's legal name. A court having jurisdiction over Carol can then compel her to reveal copies of the identity document of the pseudonymous user A_{Buv}. In this way, Alice can be held accountable if she breaks the law.

With linked identities, digital platforms are no longer a lawless zone where users can flout the law while hiding behind pseudonyms. At the same time, users who do not violate the law can enjoy a measure of protection against harassment and retaliation by using a pseudonym.

10.4 International Considerations

Players located in countries with weak legal protections for civil rights might reasonably not trust their own court system to protect them from improper unmasking. What can be done about this? One possibility is for players to choose a trusted agent in a jurisdiction that has a strong tradition of protecting civil rights. If Alice lives in Russia and Bob is domiciled in France, it may be possible for Alice to select a trusted agent Carol located in Switzerland. Bob's agent Dave might also be domiciled in France and be willing to rely on tickets provided by Carol. A court in France that issues an order against Dave to reveal the legal name of A_B (Alice's identifier with Bob) would not be engaged in a futile effort. International treaties such as the Hague Convention allow court orders from one country, such as France, to be confirmed in another country, such as Switzerland.

Telling Alice, who might be poor and live in a country with weak civil rights, that she could travel to another country is not a panacea. This solution could be impractical for many people, but it is at least an option for some people. Unfortunately, no digital identity system is going to fully cure the widespread defects in society. The proper measure of a digital identity system is whether it makes things better or worse on the margin.

Chapter 11

Privacy Preserving Record Linking

The problem of connecting records from multiple databases is called *record linking*. What we aim to achieve is called *privacy-preserving record linking (PPRL)*, where the goal is to link player records without reference to any personally identifiable information (PII) of the player. In the context of the distributed system model of this dissertation, each database is residing on a different platform.

Assume that two platforms have databases with information about players. Each platform knows the players by the hapaxes assigned to the players, which are unique per platform. For example, if Alice has accounts on the Bob and Frank platforms, Alice will be assigned the hapax A_B used with Bob and a different hapax A_F used with Frank.

What if some of Alice's information with Bob needs to be joined with some of Alice's information with Frank, and Alice consents to this? How do we accomplish it without using any other information about Alice? Assume that we cannot access Alice's legal name, her social security number, or any other identifying attribute other than the hapaxes assigned to Alice.

11.1 Related work

Previously, a system was described in Yale TR-1558 [57]. The system presented in this dissertation is a refinement of the system described in that paper, with updated terminology and mathematical functions. The contribution of that paper included a new cryptographic primitive, called a *blinding-completion pair*, which addressed the practical problem of linking deidentified records. Blinding-completion pairs provide a method for generating a multitude of pseudonyms for a player, to be used by digital platforms, and then consolidating that multitude into a single pseudonym at the database of a destination platform.

A rich body of research has addressed the problem of PPRL in the context of collecting and joining deidentified medical records. Medical database research is widespread and extremely important for the advancement of medicine. The interest in PPRL of health records goes back at least 20 years to the paper by Demuynck and De Decker [46], who propose a complicated multistakeholder protocol that uses cryptographic techniques to achieve PPRL. The idea of using pseudonym identifiers to substitute for PII linkage appears in Alhaqbani and Fidge in 2008 [32].

Several other linkage techniques have been studied in the subsequent literature. Vatsalan, Christen, and Verykios [94] give a taxonomy of PPRL techniques in a 2013 paper containing 143 references to the extensive literature on the subject! They observe that preserving the privacy of shared data, such as medical records, is a difficult problem and that existing approaches have a variety of drawbacks. For example, some systems have focused on joining records from just two sources.

Recent work aims to address one or more of the shortcomings of previous work. Camenisch and Lehmann [40] add user-auditability to pseudonym systems. The PRIMAT system [59] handles multiple sources of medical data but assumes the existence of a "trusted linkage unit (LU) [that] performs the actual linkage of the encoded records submitted".

The approach described here is different because it leverages existing universal identifiers and deidentifies them by generating two levels of pseudonyms. This results in a system

with resilience to limited breaches that is easy to understand by stakeholders. Moreover, our system does not put sensitive data in the hands of any third party. Sensitive data flows directly from a source database to a destination database. Only the identifiers related to the data are processed by third-party trusted agents.

11.2 Privacy-Preserving Linking of Sensitive Data

This section describes a new cryptographic primitive for generating identifiers that allows a destination database to link records from different data sources while preserving privacy in the face of many kinds of breaches.

11.2.1 Blinding-Completion Pairs

Let x be the identifier used by a data provider h to identify an entity. Let $f(x)$ be a one-way **alias function** that generates a pseudonym for the object referenced by x. A *blinding-completion pair* for h is a pair of one-way functions $(b_h(x), c_h(y))$ such that $f(x) = c_h(b_h(x))$ for all x, and $f(x)$ is also one-way. Like a cryptosystem $(E_h(x), D_h(y))$, the composition of the second function in the pair with the first yields the same function for all keys h. In our case, the composition is the alias function $f(x)$, which defines the alias $y = f(x)$ for x. Although neither x nor y can be recovered from $y_h = b_h(x)$, y *can* can be recovered from a single value of *any* y_h if the corresponding completion function c_h is available, since $y = c_h(y_h)$.

11.2.2 Implementation

There are several ways to implement pairs of blinding-completion functions. One way is to use cryptographic accumulators. Let $Q = \{b_1, \ldots, b_N\}$ be a set of *quasi-commutative cryptographic hash functions* [35]. They have the property that the N-way composition of these functions in any order yields the same function B. Hence, for any subset $S \subseteq Q$, the

composition of the functions in S, call it b_S, can be used as the first element of a blinding-completion pair, and the composition of $b_{(Q-S)}$ becomes the completion function c_S. The drawback of this scheme is that N must be known in advance and the time complexity of finding b_S and c_S increases with N. The system that follows uses a different scheme based on discrete logarithms.

Some Standard Number Theory

For a positive integer n, let $\mathbf{Z}_n^*$ be the multiplicative group of positive integers less than n that are relatively prime to n. A multiplicative group is a set of numbers equipped with the multiplication operation, represented by the symbol '$*$'. The size of $\mathbf{Z}_n^*$ is given by Euler's totient function $\phi(n)$. When mathematical operations are performed on elements of $\mathbf{Z}_n^*$, each operation is modulo n. In other words, if a mathematical operation results in a number greater than n, then n is subtracted repeatedly from the result until the result is less than n.

In the special case where n is a prime p, $\mathbf{Z}_p^* = \{1, \ldots, p-1\}$, so $\phi(p) = p - 1$. In addition, p has primitive roots. A number $g : g < p$ is a *primitive root* of p if every number $a \in \mathbf{Z}_p^*$ can be expressed as $a = g^k \bmod p$ for some $k \in \mathbf{Z}_p^*$. The number k is called the *discrete logarithm of a to the base g modulo p.* The computation of the discrete logarithm is believed to be impractical in polynomial time when p and g are chosen carefully.

Selecting Parameters

For the purpose of the following protocol, choose a prime $p : p = 2q + 1$, in which case q is called a *Sophie Germain prime* and p is called a *safe prime*. Such prime pairs are widely used in cryptography, so a suitable supply exists for our purposes. An estimate of the number of Sophie Germain primes less than n is $\Theta(n/(\log n)^2)$ [86, pp.123–124].

Let r, u be positive integers in $\mathbf{Z}_{\phi(p)}^* = \mathbf{Z}_{p-1}^*$, and let v be a positive integer less than $\phi(p)$ such that $r = uv \bmod \phi(p)$. Define $b_u(x) = x^u \bmod p$ and let $c_v(y) = y^v \bmod$

p. Then (b_u, c_v) is a blinding-completion pair for the alias function $f(x) = xg^r \bmod p$. According to Euler's Theorem, for $a \in \mathbf{Z}_p^*$, $a^{\phi(p)} \equiv 1 \pmod{p}$. Consequently, for $s, t \in \mathbf{Z}_{\phi(p)}^*$ and $x \in \mathbf{Z}_p^*$, if $s \equiv t \pmod{\phi(p)}$, then $x^s \equiv x^t \bmod p$. Therefore,

$$c_v(b_u(x)) \equiv c_v(x^u) \equiv (x^u)^v \equiv x^{uv} \equiv x^r \pmod{p}.$$

The parameters p, q matter both for convenience and security. To choose r from $\mathbf{Z}_{\phi(p)}^*$, we need to find a number r relatively prime to $(p-1)$. But an arbitrary $p-1$ might have many small factors, e.g. $p = 71$. However, since we choose p, q so that $p - 1 = 2q$, we know that the only factors of $p-1$ are 2 and q. Choosing a safe prime p makes it easy to find r in $\mathbf{Z}_{\phi(p)}^*$. Any odd number other than q is a suitable value for r.

We want to ensure that the blinding-completion functions we are about to define can produce a wide range of values to minimize the risk of collisions. Therefore, we should choose q to be much larger than the number of data records to be processed.

As for security, the discrete logarithm problem is hard in general, but a solution may be feasible by *Pohlig–Hellman algorithm* when $p-1$ does not have large prime factors [77]. A safe prime does not have this weakness, which is why it is "safe."

We will allow trusted agents to independently choose random values of $r, u \in \mathbf{Z}_{\phi(p)}^*$ such that $r \neq u$. We then calculate $v = ru^{-1} \bmod \phi(p)$.

Key Generation Protocol

A **constellation of identifiers**, or **constellation** for short, is a set of identifiers used by a group of players who contribute data to a platform. Each user can have a trusted agent TA_i who uses a different blinding function. The platform has a trusted agent Dave who knows the completion function to use with each participating trusted agent TA_i.

The players:

1. Alice, a player who submits a data object that needs to be identified.

2. Carol acts as Alice's trusted agent.

3. Dave acts as Bob's trusted agent.

4. Bob, a platform operating a database to receive data objects contributed by various players.

5. Trent, a trusted agent chosen by Dave to generate the blinding-completion key pairs [15].

Protocol 11.2.1. *Key Generation*

1. *Dave asks Trent to act as the dealer for a constellation sponsored by Bob. Dave provides a modulus p chosen at random, a safe prime.*

2. *Assume that Trent agrees. Trent chooses $r \in \mathbf{Z}_p^*$ at random and stores it.*

3. *For each user who asks to join the constellation, Trent chooses at random $u, v \in \mathbf{Z}_{\phi(p)}^*$ such that $uv \equiv r \pmod{\phi(p)}$. In this case, Alice wants to join the system.*

4. *Trent sends $TIC_T(< u, B >)$ to Carol. Carol saves the pair and uses u to blind the identifiers attached to the data that Alice sends Bob.*

5. *Trent sends $TIC_T(< v, C >)$ to Dave. Dave saves the pair and uses v to complete the identifiers he receives from Carol, which are then attached to the corresponding data received by Bob from Alice.*

Key Rotation

The secret key values r, u, and v should be valid for a defined epoch and then expire. Periodically, trusted agents should rotate these keys by requesting new ones from the dealer. The rotation of keys helps minimize the risk of a catastrophic breach.

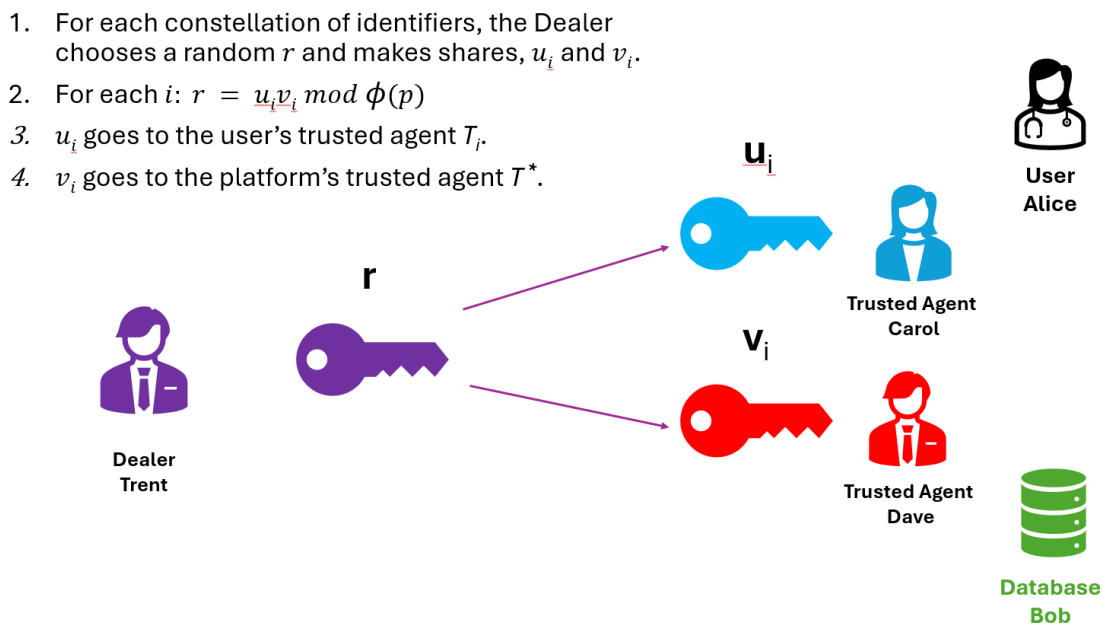

Figure 11.1: The dealer generates blinding and completion keys

Data Submission

Assume that Alice has a data object ω that she wants to send to Dave and that ω is associated with an identifier x. Assume $B_u()$ and $C_v()$ are a pair of blinding-completion functions and that $uv \equiv r \pmod{\phi(p)}$. Assume that Alice and Bob have access to an encryption function $ENCRYPT()$ and a decryption function $DECRYPT()$ to protect ω during transmission.

Protocol:

1. Alice requests a submission ticket from Carol.
2. Carol sends Alice $T_1 = TIC_C(\text{“Submission”})$.
3. Alice sends $TIC_A(< T_1, ENCRYPT(\omega) >)$ to Bob.
4. Alice sends $TIC_A(< T_1, \alpha >)$ to Carol.
5. Carol sends $TIC_C(< T_1, B_u(\alpha) >)$ to Dave.
6. Dave sends $TIC_D(< T_1, C_v(B_u(\alpha) >)$ to Bob.

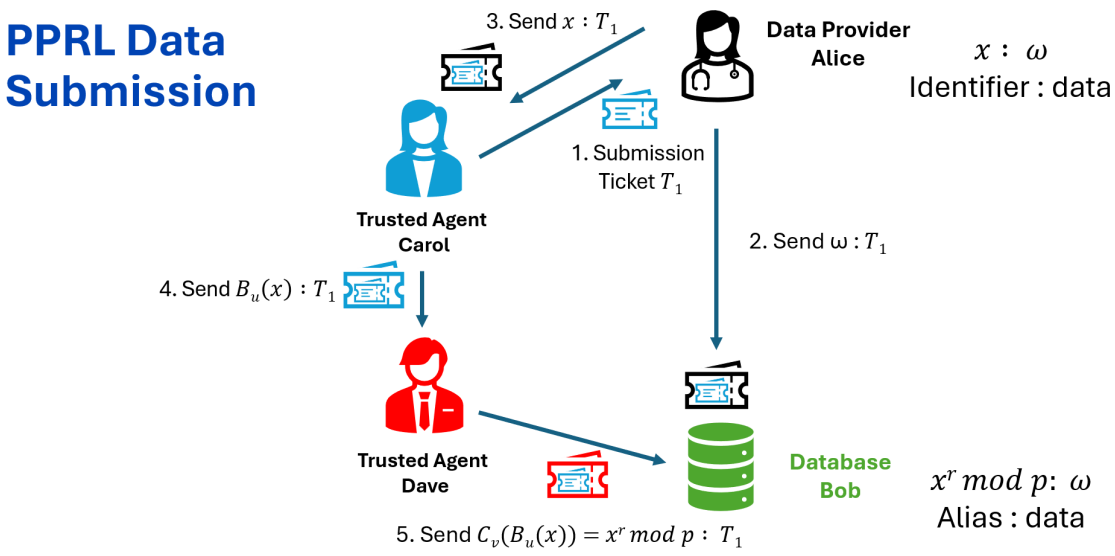

Figure 11.2: PPRL data submission protocol

7. Bob associates the alias $x^r \mod p$ with ω.

When x is a consistent identifier, such as a hash of name and date of birth, or a social security number, then $x^r \mod p$ will also be a consistent identifier. Each trusted agent in the constellation may have a different value of u, but due to the way v is chosen for the completion function, the identifier that Bob receives is always $x^r \mod p$. This protocol allows Bob to associate multiple data transmissions ω_i related to the same object referenced by x, even though Bob never receives the identifier x. Due to the impracticality of solving the discrete logarithm problem, Bob cannot calculate x given $x^r \mod p$ when suitable parameters are used.

The reason for this two-step transformation of the identifier is the property of isolation. If Carol or Dave are breached, the attacker cannot learn both x and $x^r \mod p$. This prevents an attacker from using the constellation of identifiers to learn anything more than what he could learn by breaching Alice or Bob. In other words, the constellation of identifiers does not increase the risk to privacy when there is a single breach.

Hardening the Dealer

Because the dealer is the only player who knows r, a failure of the dealer could cause a catastrophic failure. Without the dealer generating values of u and v, trusted agents cannot join the system and cannot rotate their u and v secrets.

Catastrophes can be avoided by hardening the dealer using the isolation property. A simple way to harden the dealer would be to create two independent nodes (clones) that know r and are independently capable of providing the u and v secrets on demand to the appropriate trusted agents. There is a precedent for this architecture. Internet domain names are frequently configured with two domain name service (DNS) servers, a primary and a secondary, to provide redundancy in the event that one of the servers goes down. More sophisticated architectures are also used for DNS redundancy [91].

Creating two clones of the dealer doubles the attack surface, but greatly reduces the risk of catastrophic failure. Even if one of the dealer clones suffers a data breach, the damage would not necessarily be catastrophic because the value r alone does not reveal sensitive information. An attacker who acquires r would also need to extract data records from the database to exploit r.

11.3 PPRL Prototype

A prototype PPRL system has been implemented and tested between Yale New Haven Hospital, Saint Mary's Hospital in Waterbury, CT, and the Connecticut Tumor Registry. Three pilot programs have been run to demonstrate the ability to contribute deidentified data to the Tumor Registry and join records related to the same patient. Institutional Review Board (IRB) approval has been granted to test the system on human subject data. In early 2025, testing of the system was ongoing.

Chapter 12

Sybil Account Detection

Pseudonyms are often desirable and sometimes essential in protecting users from privacy violations, harassment, or persecution [56]. In repressive countries, dissidents rely on pseudonyms for what little freedom of speech they have on social networks.

Unfortunately, unverified pseudonyms also create a risk of Sybil attacks where one or more users attempt to subvert an online voting or reputation system by casting multiple votes or making multiple comments that appear to be independent, but are controlled by the same person or persons [50]. Sybil attacks have long been a problem on the Internet, and relatively little research has been done to provide organized methods of stopping them.

Stopping Sybil attacks would provide widespread societal benefits. Examples of social problems caused by Sybil attacks include:

1. Bots, fake accounts, and troll armies run malicious influence campaigns, such as Russia's ongoing efforts to create discord online [13].

2. Fake review farms influenced $152 billion in global spending in 2022 [73]. Review farmers often take advantage of vulnerable populations desperate for work.

3. Click bots and fake traffic cost advertisers $35 billion in 2020 [60].

4. Malicious players banned from a platform return via a new account to resume toxic

behavior, leading to a game of whack-a-mole. For example, Wikipedia maintains a list of long-term abusers and documents the names of all their sockpuppet accounts (Sybil accounts) [28].

Where there is electronic voting or popularity measurement, defense against Sybil attacks is essential. The Internet has few borders (the Great Firewall of China being an exception [21]). Criminals, scams, or malicious actors in one country can easily target victims in another. International legal remedies have failed to deter Sybil attacks because:

1. The source of malicious activity may have a weak legal system. Russia, among others, hosts organized gangs that attack foreign interests. See for example, *United States of America v. Internet Research Agency LLC* (18 U.S.C. §§ 2, 371, 1349, 1028A) [14].

2. International legal action is prohibitively expensive.

3. Perpetrators often do not have recoverable assets or cannot be identified.

Therefore, digital platforms need to rely on technical defenses to mitigate Sybil attacks. Sybil attacks can be stopped using linked identities combined with privacy-preserving record linking. As explained in Chapter 10, a linked identity binds a pseudonym to a legal identifier. This relationship is known only by a trusted agent and is kept in confidence. However, the relationship could be discovered by a legal process and the associated person could be held legally responsible if they violate the law, even if they try to hide behind a pseudonym.

The effort required to establish a linked identity serves as a deterrent to large-scale Sybil attacks, such as those involving thousands of fake accounts. However, linked identities will not stop a smaller-scale attack in which a player goes to several different trusted agents and establishes several linked identities. Each trusted agent would blind the user's legal identifier using a different key value, resulting in different pseudonyms controlled by the

same person. In the linked identity protocol, each trusted agent independently chooses a random value to use as an exponent in its blinding function. This leaves digital platforms exposed to the type of Sybil attack where a banned user returns with a new identity, or a fraudster is able to run the same scam every time they set up a new account via a new trusted agent.

PPRL becomes helpful when a player tries to create multiple linked identities through different trusted agents. If a digital platform wants to resist this type of Sybil attack, the platform can set up a constellation of identifiers using blinding-completion functions. Even if a user goes to multiple trusted agents and establishes multiple accounts, the completion function will cause the multitude of pseudonyms to collapse to a single pseudonym that will be provided to the platform. The user's privacy is preserved because the platform does not learn the user's legal identity, but the digital platform can recognize when multiple pseudonyms are controlled by the same user.

12.1 Objectives

An effective method of detecting Sybil accounts would mitigate a wide array of long-standing challenges faced by digital platforms:

1. Determining if a user has more than one account on a platform.
2. Prevent a banned user from returning with a new account.
3. Prevent a user from registering with a fake identity.
4. Prevent a platform (or the public) from routinely learning the legal identity of the user through the user's identifier.
5. Allow law enforcement to discover the legal identity of a user when a court order is issued in the relevant jurisdiction.

6. Allowing governments to enact meaningful sanctions against criminal havens that produce a large volume of malicious cyber activity.

12.2 High Level Design

The key observation is that if we have linked identities, where a user's pseudonym is a blinded hash of a legal identifier, then we can use a completion function to determine the equivalence of pseudonyms that are generated from the same legal identifier. In effect, to implement a Sybil account detection feature, a platform can apply PPRL to Linked Identities.

Some digital platforms have attempted to solve the problem themselves by requiring each user to provide legal identity documents. For example, Google has built a comprehensive system for advertisers to verify their identity by submitting legal documents [3]. Smaller platforms that cannot afford to support that level of bureaucracy are placed at a disadvantage because they cannot replicate the largest platforms' verification protocols.

For users, having to send their identity documents to each platform that wishes perform a Sybil authentication creates needless redundancy. Showing legal identity documents to multiple sites is worse than inconvenient, for sending copies of sensitive documents is a major privacy risk. A network computing platform does not necessarily need to know who the user is; they just need to know that the user is a unique person and they need to verify that the person is not abusing multiple accounts on their platform. The platform needs to know that the user is not the same as a user who was previously banned for bad behavior.

Second, when legal identity documents are transmitted to multiple Internet platforms, the probability of at least one platform leaking those documents to an attacker increases. Eventually, the documents will leak and mischief will ensue. It is much better for the user to choose a trusted agent, provide the legal identity documents to that agent, and then let the agent provide a linked identity to each platform where the user needs to provide

Sybil authentication. That linked identity can be tested for equivalence to all other linked identities received by the platform to verify that the user is unique or to disclose which other accounts are controlled by the same user.

Although such a system could still be abused if a person is able to generate multiple legal identity documents, such as passports, and register them with trusted agents, such a scheme will be significantly harder to execute than merely spinning up thousands of fake accounts. When a player enters the place of business of a trusted agent and attempts to present forged or fraudulent identity documents, they could be reported to law enforcement, and might even be subject to arrest.

12.3 Implementation Sketch

This section will sketch a proposed Sybil authentication system. A constellation of identifiers is set up so that different identifiers selected by each user's trusted agent can be joined to a single common identifier by the receiving platform's trusted agent. The nomenclature used below is different from that in Chapter 10 in order to focus attention on the details of Sybil authentication and to avoid being distracted by those details that have already been explained.

In this system, Alice shows her legal identity document, such as a passport or driver's license, to her trusted agent, just as in Chapter 10. The document contains a legal identifier L_A. Then, when Alice wants to do business with Bob, Bob can perform a Sybil authentication with the help of Bob's trusted agent Dave, Alice's trusted agent Carol, and an independent dealer Trent appointed by Bob. A **Sybil authentication** to Bob is proof that a player is the same as some other player known to Bob or that the player is not the same as any other player known to Bob.

To perform Sybil authentication, the players follow Protocol 10.1.1 with the only difference being that the blinding keys u and v are not chosen at random. Instead, they are

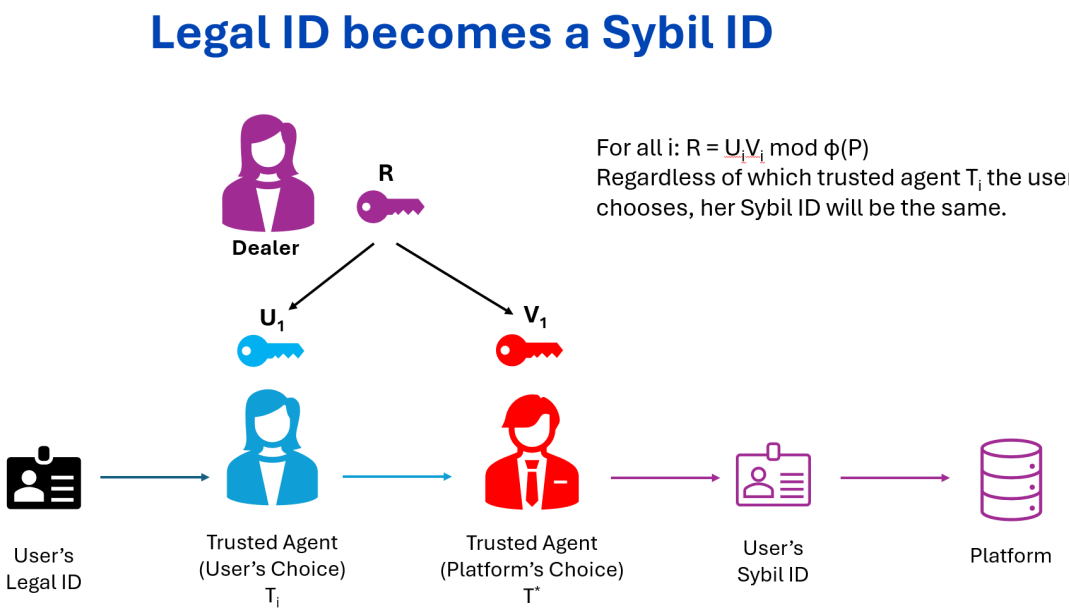

Figure 12.1: Sybil identifier generation

provisioned by the Dealer as described in Protocol 11.2.1. The resulting identifier A_{Buv} delivered to Bob is a **Sybil identifier**. The Sybil identifier is a linked identifier that has the additional property that it is invariant regardless of which trusted agent Alice chooses to work with, as long as Alice presents the same legal identifier L_A . In practice, Alice might have a driver's license with L_{A0} and a passport with L_{A1}. This is an implementation complication, not an essential problem with the protocol. Alice might be able to create a different Sybil identifier for Bob from each authoritative legal identity document she possesses. Because the number of such documents is small, the number of Sybil identities will also be small. Even a partial solution to the Sybil detection problem, one that limits a user to a handful of Sybil identifiers, would be quite useful, as will be explained in Section 12.4.

When setting up a constellation of Sybil identifiers for Bob, Dave and each of the i participating trusted agents TA_i representing players who transact with Bob, work together with Dave to choose blinding secrets u_i and v_i to perform privacy-preserving record linking as described in Chapter 11. The trusted agent of each player has a blinding secret u_i, and Bob's trusted agent has a table of pairs $\{ (v_i, TA_i) \}$ containing completion secrets and the

corresponding trusted agent name, TA_i. This table allows Dave to know which v_i to use for each transaction, depending on which TA_i submits the request.

Thus, each player can choose the trusted agent they want or even work with multiple trusted agents. Bob can still tell which players are the same and which are different from each other. For example, Bob can rely on the identifiers F_{Buv} and G_{Buv} as a signal that two players F and G presenting themselves as different players are actually the same person. If $F_{Buv} = G_{Buv}$, both F and G must have presented the same legal identification document to their trusted agents.

12.4 Problems, Claims, and Solutions

The Sybil authentication protocol sketched in Section 12.3 supports the following services that are widely applicable to modern digital platforms.

12.4.1 Identify if a Player Has More Than One Account on a Platform

When a player established a linked identity on a platform, with the legal identifier blinded and completed, the platform will receive the same consistent and persistent identifier. This means that a user could go to two different trusted agents and generate two different blinded identifiers, but because of the math used by blinding-completion functions, the completed identifier received by the platform identifier will always be the same when the original legal identifier used to set up the linked identity is the same. If a player wants to control two accounts on the platform, they could, but the platform will know that the two accounts belong to the same player, because the platform will receive the same completed identifier when performing a Sybil authentication of the player.

12.4.2 Prevent a Banned Player from Returning with a New Account

When a platform bans a player, the player's account is deactivated, and the player's identifier with the platform is added to a list of banned players. If the user returns and tries to create a new account, possibly via a different trusted agent, they still have the same legal identifier. The Sybil identifier will also be the same, and the platform will be able to apply the same enforcement measures to the new account as the original account.

If several different legal identifiers are allowed, such as license number and passport number, a user might be able to create several Sybil identities, one for each allowed type of legal identifier. However, the number of Sybil identities will be finite and small. Although not a perfect solution, Sybil authentication via linked identities and privacy-preserving record linking provides a method to prevent banned users from having endless opportunities to return and resume trouble-making like a game of "whack a mole" [93].

12.4.3 Deter a Player from Registering with a False Identity

To set up a linked identifier, the player must present legal identity documents to a trust agent, such as a bank, that operates a trusted agent. Creating a fake account with a digital platform is a minor mischief compared to walking into a local bank branch with forged documents. Trying to create a fraudulent linked identity is an entirely different scale of risk. The latter seems much more likely to result in an investigation and possibly an arrest.

12.4.4 Prevent a Platform (or the Public) from Casually Learning the Legal Identity of the Player via the Identifier

At present, platform security often focuses on passwords (long and complex) and multi-factor authentication (inconvenient and inconsistent). Digital platforms have failed to focus on preserving user privacy. Usernames are frequently reused on multiple platforms, suggesting a relationship between accounts. Many users have the same usernames on different

platforms, such as their favorite handle or email address. Reusing the same username on different platforms is bad for privacy. The username becomes a join key that allows platforms to exchange and link information about the user without consent.

Using email addresses as usernames is a common practice that is essentially a hack that provides a universally unique identifier and an easy pathway for multifactor authentication. But what if a user prefers to keep their email address secret? What if a user has multiple email addresses? What if the user changes their email address? The practice of using email addresses as usernames opens the opportunity for Sybil accounts or for the user to become confused about which email is used to log in because many users have multiple email addresses. Even worse, if a user loses access to their email address, they may lose access to accounts that depend on that address.

A linked identity supplies each platform with a consistent but unique identifier per platform for each user. Whether the user chooses to share more information (which is often inevitable), at least the username cannot be used to enable non-consensual sharing of information about the user.

As an example, Facebook requires a mobile phone number to verify users [10]. This is terrible for privacy, because Facebook can buy and sell data using the mobile phone number as a unique key for linking data to a user. Facebook can sell the online activity of users to data brokers and credit bureaus that key information by name, address, and phone number.

Users concerned with privacy should minimize the distribution of their legal identity data because platforms are frequently subject to data breaches. Platforms cannot leak information they do not have. As privacy litigation becomes more prevalent, platforms are under pressure to minimize their storage of PII. Using linked identifiers is an excellent way to minimize the amount of customer PII stored. Reducing the volume of PII protects customers from potential data breaches and can reduce platform liability if there is a breach.

Although trusted agents have access to sensitive information, there are two trusted agents involved in every transaction. If the user has taken care not to make a public disclo-

sure, an attacker cannot connect a user's identity on a platform to a legal identity without compromising two different trusted agents within the same key rotation epoch. Although it might be possible to compromise one trusted agent, breaching two in close temporal proximity would be a much lower probability occurrence.

12.4.5 Allow Law Enforcement to Discover the Legal Identity of a Player with a Court Order

Currently, law enforcement can subpoena platforms to obtain network telemetry related to a user, including IP addresses, user-agent strings, location, computer model, software version, and device identifiers. This information can provide probabilistic matching of real-life users to online activities, but often such information is only retained for a short period (typically 6 months). When covert wrongdoing occurs, critical records may be long gone before the investigation begins. Linked identifiers respect privacy and thus can be retained to provide law enforcement with a reliable identification under a court order.

12.4.6 Allow Governments to Enact Meaningful Sanctions Against Criminal Havens that Originate Malicious Traffic

Presumably, if a system for linked identities and Sybil authentication became an Internet standard, countries such as Russia would find a way to abuse the system. For example, the Internet Research Agency [14] could go to a friendly Russian trusted agent with thousands of fake passports and be able to build a troll army of Sybil accounts to attack platforms in the United States or other locations. When such activity occurs, governments can sanction violators and sever them from the digital identity infrastructure system. The governance of such systems should take care to support the rule of law and provide a mechanism for sanctioning those individuals, organizations, and countries who attempt to subvert the rule of law.

12.5 Impact on Platforms

The implementation of a standard for Sybil authentication would have consequences for platforms. Users are reticent to change how they do things, and suddenly requiring a new verification process might not be feasible. Instead, platforms could offer new features that would be available to users who establish a linked identity.

For example, a social media platform could offer to increase the reach of users who become verified with a linked identity. This would help suppress bot accounts and improve the platform's ability to sell advertising. Advertising impressions delivered to verified humans are worth more than impressions delivered to web visitors of unknown quality (which might even be bots).

Sybil authentication based on this digital identity infrastructure can be adopted by platforms with minimal effort. The platform needs to select a trusted agent and implement a relatively simple protocol to authenticate each player logging in. The platform-specific identifier received can then be compared to other identifiers previously received to determine if the player has a unique account or multiple accounts. This process can be set up with a standard website widget and standard application program interfaces (API) at the trusted agent of the platform. Tickets are signed by trusted agents who perform the transaction, thus establishing responsibility for any incorrect authentications.

Lastly, platforms could eliminate the expense and risk of maintaining their own identity verification and authentication and reduce friction from proof-of-personhood authentications such as CAPTCHAs [63]. Requesting users to send in photographs of their identity documents is neither scalable nor secure. The more places such information spreads, the greater the risk of unintended consequences and abuse.

Chapter 13

Extending RSA with a Group Public Key

This chapter describes an extension of the RSA cryptosystem that provides a single group public key (GPK) used to encrypt a message and two secret keys, each of which can decrypt the message. The group public key is efficiently calculated from two safe-to-share intermediate public keys that do not reveal the two private keys.

The RSA cryptosystem with GPK (RSA-GPK) is designed to address the requirement for restricting access to sensitive data, controlled by a single public key, that allows two (or more) independent players to each have their own secret key to access the data. The secret keys need not be transmitted. In addition, the two players do not need to communicate with each other. Each of the players must communicate once with the digital platform that holds their data to register their intermediate public keys.

RSA-GPK is designed to satisfy three requirements. First, the two players must be equally capable of decrypting a message encrypted with the GPK. Second, neither secret key may be transmitted through the communication channel. Key sharing and key escrow are not allowed. Third, the system must be backward compatible with the existing infrastructure and software that supports a single encryption key. This eliminates the trivial solution of encrypting each message using two different encryption keys because that would require extending legacy systems to support two public keys instead of one.

RSA with GPK can serve as the foundation for both a key recovery system and a method for providing alternative access to employers who need to supervise employees, guardians who need to supervise minors, spouses who want to share access, executors who need to settle the affairs of a deceased person, or courts that might order data to be provided to law enforcement.

If one secret key has been lost, the other secret key can be used to authorize a transaction to replace the lost key. Authorization can be accomplished through digital signature or other known methods. A variety of protocols for performing authorization with public key cryptography have been known since at least the 1970s [75].

For court-ordered access, called lawful access, a player can appoint a trusted agent, analogous to a corporation's registered agent, to hold the second secret key. Should a court mandate access to the player's data, a warrant could be served on the trusted agent to access the second secret key, even if the player is absent or uncooperative. Unlike previous failed proposals for lawful access, the distributed nature of RSA-GPK avoids the use of a master key, does not create a massive central honeypot of secret keys, and does not weaken encryption (when an appropriate key length is chosen).

13.1 Digital Signatures with Multiple Signers

As explained below, the GPK method can be used to allow two different parties, each with their own unique private key, to generate a digital signature that will be verified with a single public key. Group signatures [43], and ring signatures [83] are two prior methods that allow multiple users to digitally sign a message. The group public key digital signature method has different properties and is simpler to implement than both of these earlier methods.

With group signatures, "only members of the group can sign messages, the receiver can verify that it is a valid group signature but cannot discover which group member made it." Thus, group signatures have the property that they prove that a member of a group made

a signature, but do not necessarily reveal which one. “If necessary, the signature can be ’opened’ so that the person who signed the message is revealed.” Revealing the identity of the signer requires permission from the group [43]. In contrast, the GPK method does not hide who signed a message. As we will see, verifying a GPK signature reveals the intermediate public key that matches the secret key used for signing, thus distinguishing the signer from the non-signer.

Ring signatures, unlike group signatures, ”have no group managers, no setup procedures, no revocation procedures, and no coordination: Any user can choose any set of possible signers that includes himself, and sign any message using his secret key and the others’ public keys, without getting their approval or assistance. Ring signatures provide an elegant way to leak authoritative secrets in an anonymous way, to sign casual email in a way that can only be verified by its intended recipient, and to solve other problems in multiparty computations.” [83] Thus, ring signatures have distinct use cases that are not interchangeable with GPK signatures.

Unlike group signatures and ring signatures, RSA-GPK does not obfuscate the signing party, because an RSA-GPK signature reveals the signer’s intermediate public key. The unique contribution of GPK is that it allows any group member to decrypt a message encrypted with the group’s public key, or to generate a digital signature that can then be verified with the group’s public key. This is accomplished without requiring any alteration to the algorithm that performs the encryption or verifies a signature, which is extremely important to provide reverse compatibility with existing systems. RSA-GPK can be used with any system that supports authorization based on an RSA public key, and the system will work without modification.

13.1.1 Overview of the RSA Cryptosystem

The RSA cryptosystem [82] is based on the difficulty of factoring large numbers. A public key is used to encode messages and a private key is used to decode messages. RSA is

widely used to protect web traffic.

An alternative use for RSA is to generate digital signatures. To use RSA for digital signatures, the order of operations is reversed. The private key is used to sign a message and the public key is used by the recipient to verify the signature of a message [64].

To generate RSA keys, two prime numbers, p, q, are chosen at random. Let $N = pq$. In practice q must be several orders of magnitude larger than p to ensure that it is difficult to calculate Carmichael's function, $\lambda(N)$, the smallest number such that $a^{\lambda(N)} \equiv 1 \mod N$ for every integer $a > 1$ coprime to N [41]. The least common multiple function is efficiently computed using Euclid's algorithm to find the greatest common divisor and then calculate $LCM(x, y) = \frac{xy}{GCD(x, y)}$.

These mathematical relationships define the commonly used RSA cryptosystem: [64][1]

$$N = pq, \text{ the modulus} \tag{13.1}$$

$$e = 65537, \text{ the encryption exponent} \tag{13.2}$$

$$c = E_e(m) = m^e \mod N, \text{ the encryption function} \tag{13.3}$$

$$\lambda(N) = LCM(p-1, q-1), \text{ Carmichael's function} \tag{13.4}$$

$$d \equiv e^{-1} \mod \lambda(N), \text{ the decryption exponent} \tag{13.5}$$

$$m = D_e(c) = c^d \mod N, \text{ the decryption function} \tag{13.6}$$

The fundamental idea of RSA is that $\lambda(N)$ must be known to calculate the decryption exponent d from the encryption exponent e. Either finding $\lambda(N)$ directly or obtaining the factorization of an RSA modulus and then calculating $\lambda(N)$ allows an attacker to invert the encryption exponent e to recover the decryption exponent d.

Calculating the factors (p, q) given N requires a supra-polynomial time of N^2 using the fastest algorithm known. Without knowing $\lambda(N)$, it is believed to be infeasible to

[1]Hellman's cited paper uses Euler's totient $\phi(N)$. Carmichael's function $\lambda(N)$ should be used instead instead of Euler's totient because it is smaller and thus more efficient.

[2]Using the General Number Field Sieve, the fastest known factoring algorithm for large numbers [78].

calculate d given e with a classical computer[3]. RSA is believed to be secure, although a variety of attacks on RSA have been discovered that take advantage of poorly chosen parameters, deficient implementations, or side-channel attacks [38] [66]. For the purposes of this dissertation, we need not worry about these issues, which are managed through careful implementation. Software developers often use a well-established cryptography library, such as Botan [1], to avoid implementation errors that introduce vulnerabilities.

In addition, for convenience, an RSA public key in this paper is defined as the pair (N, e) and the matching private key is the triple (N, e, d). In practice, public keys are represented as more complicated structures, such as X.509 security certificates [37], which include additional fields.

13.2 Group Public Key

A group public key (GPK) uses a function F to combine two intermediate public keys, P_A and P_B to create a group public key, $P_{AB} = F(P_A, P_B)$. The group public key P_{AB} is used with an encryption function E to encrypt a message m to form a ciphertext $c = E(m, P_{AB})$. Either matching private key, S_A or S_B, can decrypt the message. If D is the decryption function, then $D(c, S_A) = D(c, S_B) = m$.

13.3 RSA with Group Public Key

Each of the two key-holding agents A and B generates an RSA key pair, as explained above.

[3] A quantum attack, Shor's algorithm, has been known for more than 30 years but has not been implemented for more than a toy problem.

A	B
$N = pq$	$K = rs$
$P_A = (N, e)$	$P_B = (K, e)$
$S_A = (N, e, d_A)$	$S_B = (K, e, d_B)$

P_A and P_B serve as intermediate public keys that may be transmitted. The group public key is $P_{AB} = (N * K, e)$. For this to work, both public keys must use the same public exponent e. This requirement is easy to satisfy in practice because RSA implementations commonly use the fourth Fermat number [42], 65537, as the public encryption exponent [38].

13.3.1 Application: Joint Access to an On-line Account

In one application, Alice could generate the intermediate public key, $P_A = (N, e)$, and the matching private key, $S_A = (N, e, d)$, on her mobile device in the device's hardware security module. Another intermediate public key $P_B = (K, e)$ could be generated by Alice's trusted agent, Bob. We assume that Bob acts in the best interests of Alice and will help her recover her account if she loses her secret key S_A. Alice and Bob could each send the intermediate keys P_A and P_B to Alice's banker, Trent. Trent could calculate a group public key $P_{AB} = (NK, e)$ and use that group key to secure Alice's account with an authentication protocol based on RSA.

The modulus values N and K should be large enough to be cryptographically secure. Cryptographically secure means that the probability of calculating d given e in polynomial time is asymptotically smaller than $1/f(x)$ for any polynomial function f, where x is the security parameter. In this case, the security parameter is the bit length of N or K. Each modulus can be chosen by a different party, and neither party needs to know the factorization of the other modulus. Both parties will be able to decrypt a message encrypted with the resulting group public key with modulus NK. Likewise, both parties can use their private key to sign a message that will be verified successfully with (NK, e).

Trent encrypts:

$$c = E(m, P_{AB}) = m^e \bmod NK \tag{13.7}$$

The Alice knowing S_A can decrypt:

$$m = D(c, S_A) = c^{d_A} \bmod N \tag{13.8}$$

The Bob knowing S_B can also decrypt:

$$m = D(c, S_B) = c^{d_B} \bmod K \tag{13.9}$$

13.3.2 RSA GPK Key Size

GPK-RSA uses more resources than classic RSA because it generates cyphertext that has twice as many bits. The size of the modulus NK is the sum of the sizes of N and K. Hence, if N and K are each $4,096$ bits, then NK and c will both be up to $8,192$ bits long. Numbers of such size are easily managed with multiprecision libraries, such as GMPY2, which enables efficient arithmetic calculations on integers, including modulo, multiplication and modular exponentiation operations [6].

13.3.3 Proof of Correctness

Why does GPK-RSA work? The key intuition is that modular exponentiation over an integer ring of size NK produces a result that is easily mapped to an integer ring of either size N or K by modular division, when N and K are relatively prime.

Lemma 1. *For three coprime integers $a > 1, N$, and K:*

$$(a \bmod NK) \bmod N = a \bmod N, \tag{13.10}$$

$$(a \bmod NK) \bmod K = a \bmod K \tag{13.11}$$

Proof: Calculate c, b_N, b_K for some $i, j \in \mathbf{Z}$:

$$c = a \bmod NK \implies c = a - jNK$$

$$b_N = a \bmod N \implies b_N = a - iN$$

$$b_K = a \bmod K \implies b_K = a - hK$$

$$c - b_N = (i - jK)N$$

$$c = b_N + (i - jK)N \implies c \bmod N = b_N$$

$$(a \bmod NK) \bmod N = a \bmod N$$

A similar proof establishes $(a \bmod NK) \bmod K = a \bmod K$.

Theorem 13.3.1 (Extracting RSA ciphertexts of intermediate public keys)**.** *It only takes one modular division to extract the ordinary RSA ciphertext produced by either intermediate public key, $P_A = (N, e)$ or $P_B = (K, e)$, from a GPK ciphertext created with the key $P_{AB} = (NK, e)$, where N and K are coprime.*

$$E(m, P_A) = E(m, P_{AB}) \bmod N, \tag{13.12}$$

$$E(m, P_B) = E(m, P_{AB}) \bmod K. \tag{13.13}$$

Proof: Apply Lemma 1 to the encryption functions for the group public key and the

intermediate public keys.

$$m^e \bmod NK \bmod N = m^e \bmod N \implies E(m, P_{NK}) \bmod N = E(m, P_N) \quad (13.14)$$

$$m^e \bmod NK \bmod K = m^e \bmod K \implies E(m, P_{NK}) \bmod K = E(m, P_K) \quad (13.15)$$

Theorem 13.3.2 (One lock, two keys)**.** *A message encrypted with the group key P_{AB} can be decrypted with either secret key S_A or S_B.*

Proof: Apply the decryption function to the equalities in 13.3.1.

$$D(E(m, P_{NK}) \bmod N, S_A) = D(E(m, P_N), S_A) = m \quad (13.16)$$

$$D(E(m, P_{NK}) \bmod K, S_B) = D(E(m, P_K), S_B) = m \quad (13.17)$$

Because the decryption functions are calculated modulo the modulus contained in each secret key, the above equations can be simplified.

$$D(E(m, P_{NK}), S_A) = D(E(m, P_N), S_A) = m \quad (13.18)$$

$$D(E(m, P_{NK}), S_B) = D(E(m, P_K), S_B) = m \quad (13.19)$$

13.3.4 A Quick Review of Sun-Tzu's Theorem

In order to prove the security of GPK, we will need to utilize Sun-Tzu's Theorem, which is also known as the Chinese Remainder Theorem.[4] It states that given a system of n modular equivalences for an integer x and n coprime moduli $m_1, ..., m_n \in \mathbb{Z}$ there exists a unique

[4]Sun-Tzu was an ancient Chinese mathematician [72].

integer R such that for $i \in [1..n]$:

$$0 \leq R \leq \prod m_i \tag{13.20}$$

$$r_i = x \bmod m_i \tag{13.21}$$

$$R = x \bmod \prod m_i \tag{13.22}$$

It follows directly that for two modular residues, such as c_N and c_K produced by two moduli N and K, a unique modular residue c_{NK} exists, modulo NK, such that:

$$c_{NK} \bmod N = c_N, \text{ and} \tag{13.23}$$

$$c_{NK} \bmod K = c_K \tag{13.24}$$

Moreover, c_{NK} can be calculated efficiently given c_N and c_K using the Extended Euclidean Algorithm.

For some $\alpha, \beta \in \mathbb{Z}$:

$$c_{NK} = c_N + \alpha * N = c_K + \beta * K \tag{13.25}$$

$$c_N + \alpha * N = c_K + \beta * K \tag{13.26}$$

$$\alpha * N - \beta * K = c_K - c_N \tag{13.27}$$

To solve the above Diophantine equation we can first solve a related Diophantine equation. We can use the extended Euclidean algorithm to calculate $\alpha', \beta' \in \mathbf{Z}$ such that $\alpha'(c_K - c_N) = \alpha$ and $\beta'(c_K - c_N) = \beta$.

$$\alpha' * N - \beta' * K \equiv 1 \mod NK \tag{13.28}$$

$$\alpha' * (c_K - c_N) * N - \beta' * (c_K - c_N) * K \equiv (c_K - c_N) \mod NK \tag{13.29}$$

Once we know α', we can calculate α, and then c_{NK}:

$$\alpha \equiv \alpha' * (c_K - c_N) \mod NK \tag{13.30}$$

$$c_{NK} \equiv c_N + \alpha * N \mod NK \tag{13.31}$$

We will utilize this method of calculation in the next section.

13.3.5 Proof of Security

Theorem 13.3.3. *An attack on RSA-GPK, having more than negligible probability of success in polynomial time as a function of modulus length, can be used to construct a novel broadcast attack against RSA.*

Corollary 1. *If RSA is secure against a broadcast attack when two copies of a message are sent after being encrypted with two public keys using different moduli and the same exponent (greater than* 2*), then RSA with Group Public Key is also secure.*

Proof: Given $c = m^e \mod NK$, assume that there exists a polynomial-time black-box function $\mathcal{D}(c, e, NK)$ that, without knowing p, q, r, or s, can, with non-negligible probability, calculate m in polynomial time, given c, e and NK as input. Let $N = pq, K = rs$, with p, q, r, s all unique primes selected at random and $m > 1$. Assume N and K are large enough to be impractical to factor using known methods and that p, q, r, s were selected with due care, as explained in the RSA overview above. Non-negligible probability means $p >= \frac{1}{f(x)}$ for infinitely many values of x where f is a polynomial function and x is the security parameter, such as the size of N or K.

Assume that the same message m is broadcast by two parties. Alice encrypts m with

$P_A = (N, e)$ and Bob encrypts m with $P_B = (K, e)$:

$$e = 65537$$

$$c_N = m^e \bmod N$$

$$c_K = m^e \bmod K$$

As shown in the section above, given c_N and c_K we can use Sun-Tzu's theorem and the Extended Euclidean algorithm to calculate:

$$c_{NK} = m^e \bmod NK$$

The extended Euclidean algorithm uses linear time as a function of the length of NK. CITE

We can then use $\mathcal{D}$ to obtain m:

$$m = \mathcal{D}(c_{NK}, e, NK)$$

If $\mathcal{D}$ exists, $\mathcal{D}(c_{NK}, e, NK)$ can successfully extract m with non-negligible probability in polynomial time, given only two distinct encryptions of the same message, including when $e > 2$. However, the most effective known broadcast attack against RSA, Hastad's attack, is impossible with fewer than e messages [38]. A broadcast attack is one where the attacker lures two or more adversaries to broadcast the same message, each encrypted with the same exponent and a different modulus. Thus, the existence of $\mathcal{D}$ implies the existence of a novel broadcast attack against RSA. Hence, if RSA is secure against a broadcast attack when only two copies of a message are encrypted with two keys using different moduli and the same exponent, then $\mathcal{D}$ does not exist.

13.3.6 Factoring Moduli with Four Prime Factors

GPK invites the interesting theoretical question of whether it is feasible to factor a modulus with four large prime factors in polynomial time as a function of the modulus. If possible, this method could also be used to factor in polynomial time $N = pq$, with p and q large primes.

Theorem 13.3.4. *Assume there exists a black-box polynomial time function that can extract a single prime factor from a composite number that has exactly four prime factors chosen at random $\mathcal{D}(pqrs) = p$, with non-negligible probability. Then $\mathcal{D}$ also has a non-negligible probability of successfully factoring an RSA number with two prime factors chosen at random.*

Proof: Suppose that we want to factor $N = pq$, the product of two primes chosen at random. We choose two random primes, r and s in the interval $[2, N/2]$ such that $r \neq s$, $GCD(r, N) = 1$, and $GCD(s, N) = 1$.[5] We calculate $t = \mathcal{D}(Nrs)$. With probability $\frac{1}{2}$, we find $t \in \{ p, q \}$. Therefore, if a composite number with four prime factors could be factored in polynomial time with non-negligible probability, there is a non-negligible probability of factoring a RSA number pq in polynomial time when p and q have been chosen at random.

13.3.7 RSA-GPK Signatures

To construct a signature scheme, we use each participant's secret key to sign and we use the group public key to verify. We start by constructing a group public key G_{ab}:

1. Alice has intermediate public key $P_a = (N, e)$
2. Bob has intermediate public key $P_b = (K, e)$
3. The group public key is $G_{ab} = (NK, e)$

[5] If we get extraordinarily lucky and guess a factor of N, our task is complete.

Alice and Bob have each chosen their moduli independently at random, resulting in a negligible probability that they have chosen the same secret key:

1. Alice has $d_a \equiv e^{-1} \mod \lambda(N)$
2. Bob has $d_b \equiv e^{-1} \mod \lambda(K)$

The signature process utilizes either a secret key and a suitable hash function H, such that $1 < H(m) < NK$:

1. Calculate $s = H(m)$
2. Alice signs $Sig_A = s^{d_a} \mod N$, or
3. Bob signs $Sig_B = s^{d_b} \mod K$

The verifier receives $Sig \in \{ Sig_A, Sig_B \}$, calculates the hash of the message, and proceeds to verify the signature:

1. $s = H(m)$
2. $V = Sig^e \mod NK$
3. If $Y = GCD(V - s, NK) > 1$, the signature is valid.

Theorem 13.3.5. *A group public key signature does not provide deniability.*

Assume that Alice or Bob created a signature Sig_X.

Lemma 2. *The verifier can determine who created Sig_X because the verification value Y will either be a multiple of N or K, with all but negligible probability, depending upon whether Alice or Bob created the signature.*

Proof: Assume without loss of generality that the verifier receives Sig_A.

$$V = {Sig_A}^e \bmod G_A B \tag{13.32}$$

$$V = \left(s^{d_a} \bmod N\right)^e \bmod NK \tag{13.33}$$

$$V = s + \delta N, \text{for some } \delta \in \mathbb{Z} \tag{13.34}$$

$$Y = GCD(V - s, NK) \tag{13.35}$$

$$Y = GCD(\delta N, NK) \tag{13.36}$$

$$Y = \gamma N \tag{13.37}$$

But what if $K|\gamma$? That case is equivalent to $Y = NK$, which can only occur when $V = s$. In that case $Sig_A^e \bmod NK = s$, which implies that $d_A e \equiv 1 \mod \lambda(NK)$. This can only happen when N and K are not coprime. The probability of this happening is negligible because Alice and Bob chose N and K independently at random.

13.3.8 GPK Protocol Description

The GPK protocol describes how a user Alice can designate a trusted agent Trent and set up a Group Public Key broadcast by a service.

Players:

- A - The client Alice
- T - Alice's trusted agent, Trent
- S - Service, who advertises the group public key for Alice's Group

Secrets:

- $p, q, N = pq, d_A = e^{-1} \bmod \lambda(N)$
- $r, s, K = rs, d_T = e^{-1} \bmod \lambda(K)$

Public:

- $M = KN$ the group public key
- e the encryption exponent

Key Generation Protocol (sketch):

Assume that A and T have an existing trust relationship that allows them to identify and authenticate each other. Likewise, assume the same for A and S.

1. Pair-wise authenticated handshakes (A, S) and (A, T) establish secure and authenticated channels between the two pairs.
2. A asks T to join her group with S. Assume T agrees.
3. A generates RSA primes p, q, intermediate key (N, e), and computes d_A,
4. T generates RSA primes r, s, intermediate key (K, e), and computes d_T.
5. A sends (N, T) to S, indicating that she wishes to add T to her group as a trusted agent.
6. T sends (K, A) to S, indicating that he agrees to join the group of A.
7. S computes $M_A = NK$.
8. S puts $(Alice, (M_A, e))$ in the public key database. S also remembers who T is, but it should be kept in a secure store so that it is difficult for an attacker to learn who Alice's trust agent is.
9. A keeps private key (N, e, d_A) in a local key store.
10. T keeps private key (K, e, d_K) in a highly secure and high latency escrow database.

Now, we have to ask what each player knows and what can an attacker do after compromising one of the players?

Alice knows N and d_A. During key generation, she also knew p, q, and $\lambda(N)$, but these should be securely erased after generating d_A. Trust node Trent knows K and d_T. During key generation, he also knew r, s, and $\lambda(K)$, but these should be securely erased after generating d_T.

The service S learns N, K, and $M_A = NK$. S retains M_A in the public database. The only confidential information that S keeps is the identity of T, Alice's trust node, and perhaps some data used to authenticate T, such as a public key. If A is compromised, the attacker can read incoming encrypted messages and can compute K, but this information will not help the attacker learn d_T. Similarly, if T is compromised, the attacker can read incoming encrypted messages and compute N, but this information will not help the attacker learn d_A. Moreover, since T has placed d_A in escrow storage, a compromise of T by itself might be insufficient to recover K and d_T. For example, the escrow database might itself be encrypted by a key that lives offline and is only used when responding to valid key recovery requests.

13.4 Features and Limitations of GPK

GPK has four important security properties.

1. The ability to generate a second secret key alleviates the need to copy and transmit a secret key to a secure storage location or key escrow. Instead, the second secret key can be generated using hardware held by a trusted agent, partner, or spouse. The secret never needs to leave the secure enclave where it is generated.

2. The parties do not need to perform a key exchange protocol such as Diffie-Hellman [48]. Key exchange may appear simple, but implementation can be complicated [31]. In addition, secure key exchange requires both parties to authenticate the other party

to ensure that they are not exchanging keys with an impostor. This requirement is far from simple when the keys reside on a mobile phone controlled by a consumer, which is only periodically connected to the Internet and frequently jumps to new IP addresses. In addition, key exchange has only been reliably implemented between two parties [19].

3. GPK can be generalized to groups of three or more. For example, if three players have public keys $(N, e), (K, e)$, and (M, e), where e is the common encryption exponent and N, K, and M are the moduli, then (NKM, e) serves as a GPK.

4. GPK distributes the secrets among independent key holders that are not controlled by a centralized platform, thus preserving the property of isolation.

GPK has at least two limitations that may be relevant to applications of the method. First, the size of the group public key grows linearly in the number of group members, so it is best suited for small groups such as two or a handful. Second, GPK signatures are not anonymous. Each GPK signature emits the public key of the signer. A public key acts as a pseudonym that can link records of different transactions signed with the same private key. GPK should not be used for applications that require deniability.

13.5 Future Work

Future work includes determining which widely used cryptosystems can work with the GPK method. It is possible that ElGamal and Elliptic Curve methods would also be amenable to GPK, because these are all based on modular exponentiation. It is an open question whether GPK, could work with post-quantum methods, because these systems have been designed to avoid the hidden subgroup problem (which subsumes the modular logarithm problem). Hidden subgroups can be found in polynomial time by Shor's algorithm running on a quantum computer.

Future work also includes designing and building key recovery systems and lawful access systems based on the RSA-GPK method. For example, a user could designate a trusted agent to hold a second key for each account belonging to the user. In the event the user loses one or more of their keys, the trusted agent could still authenticate to access the account and help the user replace their lost key with a new one.

Such a key recovery system can also facilitate access in a variety of edge case scenarios. For example, a corporation may have legal reasons to access data held by an employee. A parent or guardian may need access to supervise the online activities of a minor child. An executor may need access to the accounts of a deceased person.

A court, upon a showing of good cause, may order cryptographically hidden data to be revealed. If both a user and their trusted agent could access an account, if law enforcement obtained a court order to search the account, this order could be served on the trusted agent, even if the user cannot be served papers or refuses to comply. In this scenario, trusted agents could be regulated to ensure legal behavior.

Chapter 14

Key Recovery and Lawful Access

14.1 Introduction

As the Internet, cloud computing, and e-Commerce have become essential to society, the risk of people losing access to their data and digitally managed assets has multiplied. To access their data and assets, people must maintain possession of their authentication secrets, which could be passwords, cryptographic keys, or hardware authentication devices, such as those based on biometrics. With the growing diversity of multifactor authentication systems, the management of secrets and sensitive equipment has become an increasing burden for everyone.

The risk of key loss is of a similar magnitude as the risk of being hacked. Although the incremental risk of key loss may be small, over time, it becomes likely that someone will lose one or more of their secret keys. Hardware fails or goes missing. In the UK, a man claims to have thrown out a hard drive containing the key to a wallet containing $700 million Bitcoin [88]. Backups can fail because people usually do not try their disaster recovery process until they face an actual emergency. In addition, everyone eventually dies, leaving an estate that needs to be settled. In the Digital Age, estates may contain valuable data and digital assets.

Executors, trustees, and probate courts should be able to recover and manage the decedent's assets, even when an authentication secret might be unavailable. In addition, parents may need to monitor the activities of minor children on digital platforms. Governments should be able to conduct targeted investigations when they have evidence of a crime, often called "lawful access." However, lawful access should be designed in a way that does not facilitate mass surveillance. Methods for managing "key recovery", an umbrella term for resetting or retrieving authentication secrets, can provide these features, but existing methods have failed to establish a satisfactory balance between the competing requirements of convenience, security and lawful access [30].

According to the US Department of Justice, "Law enforcement is increasingly facing challenges due to the phenomenon of 'warrant-proof' encryption. Service providers, device manufacturers, and application developers are deploying products and services with encryption that can only be decrypted by the end user or customer. Because of warrant-proof encryption, the government often cannot obtain the electronic evidence and intelligence necessary to investigate and prosecute threats to public safety and national security, even with a warrant or court order. This provides a 'lawless space' that criminals, terrorists, and other bad actors can exploit for their nefarious ends" [17].

Having two keys that can independently open a lock is a common practice in everyday life. Many households have a joint checking account where either one of two spouses can sign a check. Every new car comes with two keys. These solutions evolved because they are practical and socially beneficial. The cryptographic method described in Chapter 13, called a group public key (GPK), reflects the patterns of natural human behavior. We are sociable, but error prone, often losing our things. We usually have someone we trust who can help us when needed. We often work in teams of two or more. Consequently, if we need to keep a secret key, we are better off with two because we will likely lose one or the other. Moreover, it should be possible to provision or revoke the secrets independently of each other.

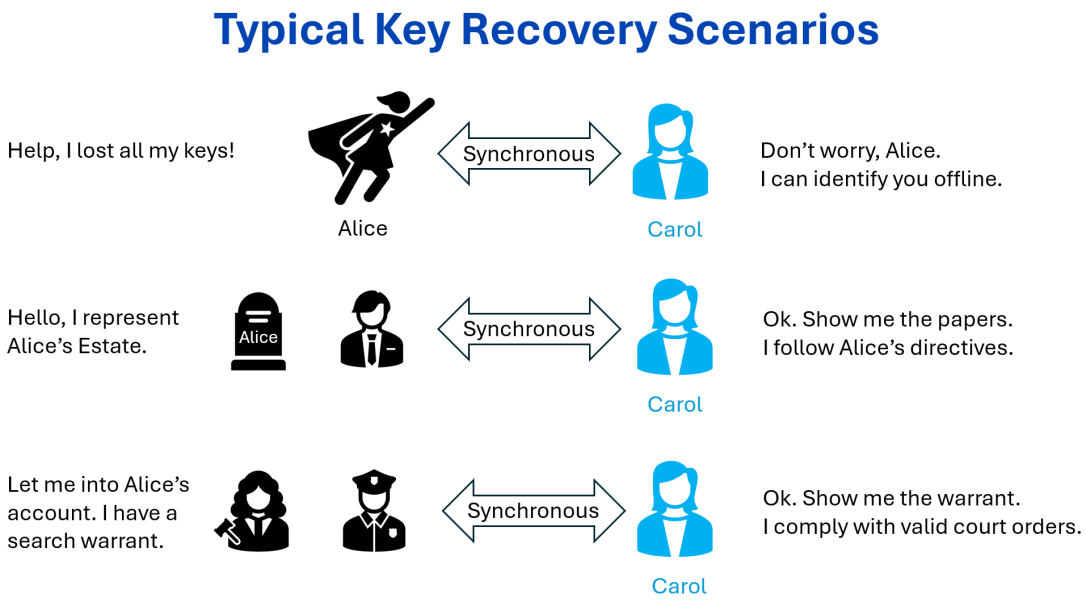

Figure 14.1: Scenarios where Alice's trusted agent Carol can provide key recovery

Based on GPK, linked identities, or both, this chapter will sketch a method to provide lawful access while maintaining security without a significant increase in risk. In this context, lawful access could mean restoring access to a user who loses all of her keys, providing access to person lawfully authorized to manage the affairs of a user who dies and their keys are no longer available, or unlocking accounts or devices to comply with a court issued search warrant.

The key recovery method sketched in this chapter provides lawful access to digital platform accounts, as well as to hardware with digital locks, without compromising security against third party attacks.

By providing two (or more) secret keys with equal authority that work with the same identity commitment (a verification key), the group public key (GPK) method reduces total risks to the user in practice. "In practice" is an important caveat because at first glance it might appear that having duplicate secrets would increase the attack surface. However, in practice, online security and authentication systems almost always have a backup method or means of resetting the secret. This backup, or secondary method of access, has already expanded the attack surface. With GPK and trusted agents, players can manage that ex-

panded attack surface in a better way.

14.1.1 Risk analysis

Two keys are much better than one key because the probability of losing two independent keys at the same time is much lower. If the probability of key loss is ρ per time interval, the probability of losing both keys in the same interval is ρ^2. Presumably, when one key of two keys is lost, the other key will allow a player to replace the lost key in the next time interval.

If the probability of a key being exposed is ζ per time interval, then the probability of having at least one of two keys exposed is at most 2ζ per time interval. If ζ is negligible, such that $\zeta < \frac{1}{g(x)}$ for all polynomial functions g of the security parameter x when x is sufficiently large, then 2ζ is also negligible.

Although the risk of key theft ζ is negligible with one key and remains negligible at 2ζ with two keys, the risk of key loss ρ geometrically decreases to ρ^2 with the addition of a second key. The risk of breach remains on the same order of magnitude, whereas the risk of lockout decreases by many orders of magnitude. This trade-off would benefit players in most circumstances.

14.2 Key Recovery and Lawful Access Protocols

Currently, digital platforms have non-uniform methods to restore lost keys. Having a different process for each platform is undesirable because a user who loses their keys or key storage device needs to perform a bespoke key recovery process for each platform. A uniform and non-repetitive process would be a great improvement in usability and security.

For platforms that use passwords, the typical key recovery method is to request a password reset by email. The problem with designing the protocol this way is that too much trust is placed in the email address. If the email account is compromised, it becomes a

single point of failure that could lead to catastrophic breach, from the perspective of the player, of multiple platform accounts at once.

14.2.1 Recovery Via Trusted Agents

When platforms use a federated identity system, such as the one outlined in chapter 9, they no longer are responsible for key recovery. Instead, the user can solve any key loss problem with their trusted agent. This could be as simple as canceling the lost key and provisioning a new one using the protocols already described in Chapter 7.

What if a player loses all their keys at once, such as if they are robbed of all their devices? If the player has established a linked identity, they can visit the physical office of their trusted agent. They can present legal identification. They can provide evidence that they are the same person as the photograph that they previously provided to their trusted agent. They can provide a fingerprint or retinal scan. They can provide a signature to show that it is the same as the one in the file. They can sign an affidavit under penalty of perjury that they are who they claim to be. They can summon witnesses. Although not risk-free, a combination of these methods would be practical. Importantly, such a process mirrors what is already the existing practice in offline business and, thus, will be readily accepted in society and in court.

If someone dies suddenly, they may never have the opportunity to share their passwords. If a player dies, their executor can present a death certificate and the will to a the deceased's trusted agent and petition to gain access to the deceased's accounts. The key recovery process facilitates the restoration of access after a catastrophe, including when the account owner dies. Trying to present such proof of death and inheritance to each on-line account would be extremely inefficient and might be infeasible in many cases. It is much better to be able to present that proof to a trusted agent as part of the key recovery process.

This key recovery process also serves to provide lawful access. Under a court order, law enforcement can ask a trusted agent to add another key, one controlled by law enforcement,

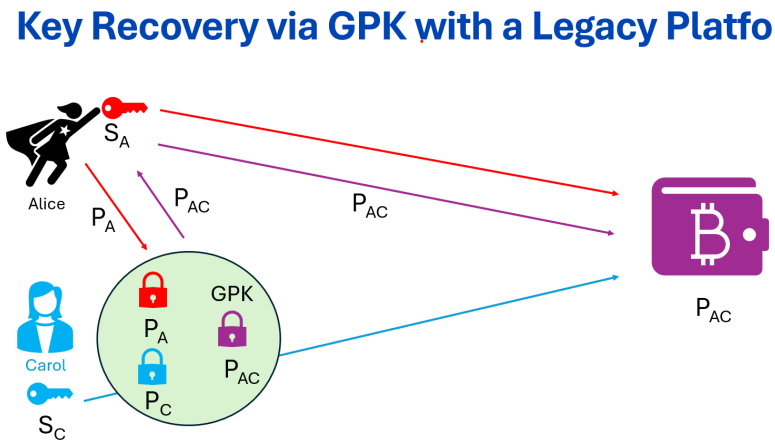

Figure 14.2: GPK key recovery with a legacy platform

to a player's key set, and then use that key to gain access to whichever of the user's accounts is authorized by the court order.

14.2.2 GPK is Compatible with Legacy Systems

Why is GPK helpful if players can perform key recovery through trusted agents? One of the important factors that accelerates the adoption of new technologies is whether these technologies are compatible with previously adopted ideas [84, pp. 243-246]. For example, if players are used to providing a passkey to access an online service, GPK can produce a passkey that works with two different secret keys. The GPK passkey can be provided to the online service without changing the way that service works. GPK is compatible with existing solutions that rely on public-key cryptography for authentication because a GPK public key looks like a standard RSA public key. A system that expects an RSA public key to be supplied when pinning an identity commitment can also accept a GPK key without changing even a single line of code. Both the standard RSA and the GPK encryption and decryption functions work exactly the same way. The only difference is that with GPK there can be two independently generated and held secret keys that can successfully solve a puzzle generated with the GPK key.

When a platform (or hardware device) does not participate in the UNS federated identity

system, a player can work with their trusted agent to generate a GPK and provide it as their identity commitment. As a result, both the player and the trusted agent have a secret key that allows them to perform an authentication transaction. If the player loses their secret key, the trusted agent can use its secret key to unlock the account and help the player register a new verification key.

14.3 Public Policy Implications

During the 1990s, the Crypto Wars ignited a public policy debate about whether government agencies should have the ability to decrypt messages to facilitate law enforcement and help deter terrorism. Lawful access means that governments can access a decryption key based on a court order, even if the user refuses to cooperate. Several schemes were mooted based upon key escrow, foremost the Clipper Chip and Skipjack cipher [36].

In 1998 these ideas were condemned by top cryptography experts as bad for security due to the difficulty in implementing key recovery in a secure manner [29]. In addition to the severe vulnerabilities that were identified in the Clipper Chip, an important general criticism was that any central repository of keys would be a giant honeypot that would attract attacks, with the potential for great damage from any breach. Defending such a honey pot, especially against nation-state actors, would be nearly impossible.

The Crypto Wars continue under new guises. In 2015, many of the same authors of the 1998 article co-authored a new report titled *Keys Under Doormats: Mandating insecurity by requiring government access to all data and communications* [30]. By 2015, the same challenges to secure key recovery remained and the stakes were even higher due to the increasing importance of cryptography for electronic commerce and civil society.

Ten years later, we still have no Internet standard for key recovery and lawful access. Governments continue to push bad ideas that are likely to compromise user privacy, such as *Joint Declaration of the European Police Chiefs* [24]. To avoid such bad ideas, this chapter

aims to provide a method of lawful access that would support the rule of law without undermining the security benefits provided by encryption.

14.3.1 Lawful Access and Exceptional Access

There is a subtle difference between lawful access and exceptional access. In this chapter, **lawful access** means that law enforcement can obtain a decryption key after receiving a court order. Many past discussions have used the phrase **exceptional access** in situations where law enforcement would already possess a master key or hold keys in escrow and thus be able to decrypt private communications without even obtaining a court order [70].

Exceptional access is a contentious subject because such access has historically been subject to poor supervision and frequent abuse. The opposite extreme, where law enforcement is generally unable to decrypt messages, even with a warrant, risks creating a lawless online realm where crime flourishes. Over time, such lawless realms pose a grave risk to civil society.

The conventional wisdom has been that lawful access is incompatible with security. The Electronic Frontier Foundation says, "...law enforcement wants a way to bypass encryption without making it easier for these bad actors. Unfortunately, a multitude of expert cryptographers have concluded that there's no feasible way to do this." [20] The EFF statement is not a proof of impossibility. Rather, it accurately implies that the cryptographers did not know a way for security and lawful access to co-exist. As science advances, things that were once infeasible become practical. GPK is a meaningful step toward enabling lawful access without compromising the security of encryption.

A master key is highly problematic because a master key would become a target for determined attackers, whereby a single breach could lead to catastrophic failure. Even if the chance of an attack is small, over time the risk accumulates. Attacks might be mathematically infeasible with current technology (e.g., factoring the modulus of an RSA public key), but insider risk and physical security have no such guarantees. GPK provides a way to

construct a lawful access system that does not use master keys, thus preserving the benefits of isolation.

What if GPK is used for lawful access? GPK does not aim to protect against a lawless government that is unwilling to submit to the authority of legitimate courts. This social-legal problem is outside the scope of a technical solution. People will inevitably suffer the consequences of poor governance. The government can exercise dominion over a person via physical custody to exert pressure until the target reveals his or her secrets.

Each person should be assigned different secret keys and allowed to choose an agent they trust to hold their second key. By fragmenting the universe of key holders, under the principle of isolation, such a system avoids the creation of a central honeypot and prevents catastrophic failures, even if trusted agents suffer an occasional breach.

An attacker who compromises trusted agents in hopes of creating a portfolio of secret keys that can be used later to target whomever they want will face two impediments. First, trusted agents can frequently rotate their keys, invalidating previously stolen keys. Second, the identity of a target's trusted agent is not public information, so that information would also need to be acquired contemporaneously to execute an attack against a specific target.

14.3.2 Forward Secrecy

A GPK-based lawful access system does not necessarily provide access to all encrypted communications. As commonly implemented, the setup of secure communication channels involves two steps. First, communicating players need to authenticate each other to know that they are connecting with the intended player. This step commonly uses an asymmetric cryptographic protocol in which each player has a public-private key pair. In the second step, players generate a shared symmetric encryption key, called a **session key** that allows efficient encryption and decryption. This second step is necessary because asymmetric cryptography is less convenient and less efficient than symmetric cryptography. The session key is used for one conversation (session) and then is discarded.

A simple and less secure method of key generation called **key wrapping** involves one player choosing a symmetric session key at random, encrypting it with the asymmetric public key of the other player, and then transmitting the wrapped key through an insecure communication channel where eavesdroppers may be listening.

A better and more sophisticated method of key generation requires players to run a key generation protocol such as Diffie-Hellman to generate the symmetric session key that allows efficient encryption and decryption [48]. Diffie-Hellman assumes that the endpoints have already authenticated each other's identity, which explains why the first step of the secure communication channel setup protocol is required.

The Diffie-Hellman protocol between two players, Alice and Bob, involves several steps. These steps are simplified for comprehension. An actual implementation may be more complex:

1. Alice and Bob agree on a modulus p that is a safe prime, and a base value g that is a generator of the modular multiplicative group $\mathbf{Z}_p^*$.
2. Alice chooses a random value $a : 1 < a < p - 1$
3. Bob chooses a random value $b : 1 < b < p - 1$
4. Alice sends Bob $g^a \bmod p$
5. Bob sends Alice $g^b \bmod p$
6. Alice calculates the session key $s = (g^b)^a \mod p$
7. Bob calculates the session key $s = (g^a)^b \mod p$

An eavesdropper Eve cannot calculate s because she does not know a or b. Eve may know the parameters g and p and the intermediate values $g^a \bmod p$ and $g^b \bmod p$. Given all this information, subject to the hypothesis that the discrete logarithm problem (DLP) cannot be solved in polynomial time when appropriate parameters have been selected, Eve cannot calculate s in polynomial time.

Because Diffie-Hellman does not involve key wrapping, a subsequent lawful access request that reveals one or both of the players' asymmetric secret keys will be insufficient to recover the symmetric session key. This property is called **forward secrecy**. Thus, forward secrecy means that even if an eavesdropper obtains one of the player's secret keys, the eavesdropper will not be able to decrypt prior conversations.

Lawful access based on GPK would allow online accounts to be unlocked, hardware to be unlocked, and a law enforcement officer to impersonate a player. By impersonating a player, law enforcement may be able to gather valuable evidence from an encrypted conversation that begins after access has been obtained. In addition, if law enforcement records an encrypted conversation that involved the transmission of a wrapped key, law enforcement may be able to decrypt that conversation.

The immediate question is whether the GPK method is sufficient in the presence of forward secrecy. This philosophical question should be viewed in the context of how people communicated before the advent of electronic communications. If Alice sent Bob a letter and Bob saved it, Bob might have had to provide that letter to law enforcement under due process of law. If Alice met Bob in a noisy pub and they have a conversation, the government would have had trouble recovering that conversation. In that context, forward secrecy is not a problem. It is merely a continuation of what has always been.

14.3.3 Illustrative Use Case

Suppose Eve and Mallory are conspiring to commit serious crimes. Leo, the law enforcement officer, conducts an investigation and uncovers evidence of ongoing criminal activity involving Eve and Mallory. He obtains a wire tap authorization and intercepts Eve and Mallory's encrypted messages, but he cannot decrypt those messages yet.

Suppose that Eve and Mallory are using a messaging app that provides encryption. The app may require its customers to nominate a trusted agent who can also decrypt their communications when served a court order. If Leo gets such an order, based upon probable

cause, he could go to Eve's agent or Mallory's agent and compel them to authenticate Leo so that he can impersonate Eve or Mallory in future conversations. Leo may also find data stored on the messaging app account after logging in. Additionally, if the messaging app does not provide forward secrecy, Leo may be able to recover symmetric encryption keys that allow him to decrypt previously intercepted encrypted messages between Eve and Mallory.

Why is this different from simply compelling Eve or Mallory to decrypt their own messages or unlock their own accounts for inspection by Leo? The messaging app operator might not know their legal names or addresses. In addition to being hard to find, Eve and Mallory could destroy their keys if they discover that they are the targets of an investigation. A trusted agent, such as a lawyer or bank, would be unable to hide from a process server and would refrain from spoliation of evidence. The trusted agent has too much exposure and too much at stake to take such an extreme risk.

Trusted agents could be licensed, post bonds, and be subject to a code of conduct. A trusted agent could be analogous to a registered corporate agent that has a registered address and reliably receives legal papers. A trusted agent could also act as a fiduciary on behalf of their clients and resist improper attempts to breach a client's privacy. They can go to court and contest an improper request for the client's confidential information.

Chapter 15

Conclusions

Security and privacy depend on both cryptography and infrastructure, including tickets, trusted agents, and hardware devices such as routers, firewalls, and hardware security modules. Although we have made considerable progress in the field of cryptography, relatively less effort has been put into developing the infrastructure needed to secure the Internet. The result is a patchwork of authentication systems that leaves users confused, insecure, and lacking privacy. Instead, we need a uniform system for communicating verifiable assertions, one that maintains the principles of Just Enough Trust.

A federated identity system called UNS has been prototyped and is available for educational and experimental use. UNS uses trusted agents to build a chain of trust between users and digital platforms. User's can login to the Internet and then reveal as much information as they like to each site they visit. Tickets are issued and saved to provide an audit trail in case of future disputes.

A method using blinding/completion functions has been demonstrated to perform privacy-preserving record linking. Medical database research is crucial for discovering new treatments, but such research has been hampered by the need to maintain the privacy of patient data. A system to reliably link deidentified medical records has long been needed. Ongoing testing will evaluate the viability of a medical data PPRL system under realistic conditions.

Linked identities provide a way to maintain a user's privacy up to the moment a court orders them to be identified. The potential of being exposed serves as a deterrent to malicious activities. Therefore, linked identities help prevent the formation of lawless realms. Future work should consider what to do when a user cannot be held accountable via a linked identity. Under what circumstances should the platform be held accountable instead? Should laws (such as Section 230 [9]) be based on the principle that platforms are immune from liability for user-provided content only when the content can be attributed to a specific person? If a platform cannot prove that content was provided by a legally identifiable user, should the platform be responsible for that content as if it were their own? Linked identities also provide a mechanism to verify user attributes, such as age or citizenship, without revealing additional information. This feature is strongly pro-privacy.

Sybil attacks cause widespread chaos on digital platforms, including ad fraud, review stacking, and systematic distortion of public opinion. Platform-specific solutions are inconsistent and have not stopped the problem. A system for reliable Sybil account detection and defense has been proposed and should be tested. With the support of digital identity infrastructure, Sybil account detection and defense is a tractable problem.

A system for composing two RSA public keys to form a Group Public Key has been demonstrated. GPK has been proven to be as secure as RSA, although GPK requires a key twice as large. Methods for encryption-decryption and digital signatures with GPK have been detailed. An exemplary GPK protocol demonstrates how to implement the GPK method. Such an implementation could form the basis for a robust key-recovery system and a robust method for multiple parties to access an online account, including employer supervision of employees, parental supervision of minors, and lawful access when ordered by a court.

An identity system is only as strong as its key recovery mechanism. There are many good reasons to encourage the adoption of a uniform key recovery infrastructure, including user convenience, the avoidance of single points of failure, and the provision of lawful

access. The GPK method provides a way to implement key recovery in legacy systems without requiring a change in the software. Trusted agents also provide a sophisticated method of key recovery that can accommodate a wide range of use cases.

Together, these digital identity services demonstrate that lawful access and rule of law can co-exist with robust privacy and security. Digital platforms can no longer justify their insecurity and inconvenience by claiming that digital identity is too hard.

Bibliography

[1] Botan. https://botan.randombit.net. [Accessed 17-01-2025].

[2] Civil Procedure - State Laws. https://www.law.cornell.edu/wex/table_civil_procedure. [Accessed 13-03-2025].

[3] Document requirements for advertiser verification - United States - Advertising Policies Help. https://support.google.com/adspolicy/answer/9872280. [Accessed 22-01-2025].

[4] ID-card - e-Estonia. https://e-estonia.com/solutions/estonian-e-identity/id-card/. [Accessed 17-02-2025].

[5] Login or Create an Account. https://myaccount.nytimes.com/auth/enter-email. [Accessed 16-02-2025].

[6] Overview; gmpy2 2.2.1 documentation. https://gmpy2.readthedocs.io/en/latest/overview.html. [Accessed 20-08-2024].

[7] Sanctions List Search. https://sanctionssearch.ofac.treas.gov. [Accessed 14-03-2025].

[8] Sign In or Create a New Account. https://sa.www4.irs.gov/sso/. [Accessed 16-02-2025].

[9] What you should know about Section 230, the rule that shaped today's internet. https://www.pbs.org. [Accessed 15-01-2025].

[10] Why am I being asked to add my mobile phone number to my account? — Facebook Help Center. https://www.facebook.com/help/1137953062904148/. [Accessed 17-02-2025].

[11] Blog: "This Is a Watchbird Watching YOU". https://libwww.freelibrary.org/blog/post/1404, 2011. [Accessed 14-03-2025].

[12] Ross Ulbricht, The Creator And Owner Of The "Silk Road" Website, Found Guilty In Manhattan Federal Court On All Counts. https://www.justice.gov/, 2015. [Accessed 01-02-2025].

[13] Exposing Russia's Effort to Sow Discord Online: The Internet Research Agency and Advertisements — Permanent Select Committee on Intelligence Democrats. https://democrats-intelligence.house.gov/social-media-content/, 2018. [Accessed 20-01-2025].

[14] Grand Jury Indicts Thirteen Russian Individuals and Three Russian Companies for Scheme to Interfere in the United States Political System. https://www.justice.gov, 2018. [Accessed 22-01-2025].

[15] Conversation with Ben Fisch. Personal communication, Oct 2022.

[16] Evidence. https://www.law.cornell.edu/wex/evidence, 2022. [Accessed 02-02-2025].

[17] Lawful Access. https://www.justice.gov/olp/lawful-access, 2022. [Accessed 15-03-2025].

[18] The right to privacy in the digital age. https://digitallibrary.un.org/record/3999709?v=pdf, 2022. [Accessed 25-02-2025].

[19] Conversation with Aikaterina Sotiraki. Personal communication, Oct 2023.

[20] Encryption and "Exceptional Access". EFF.org, Nov 2023.

[21] The Chinese Firewall - Internet Society. https://www.internetsociety.org/resources/internet-fragmentation/the-chinese-firewall/, 2023. [Accessed 22-01-2025].

[22] An Update on Sharing. https://about.netflix.com/en/news/an-update-on-sharing, 2023. [Accessed 22-02-2025].

[23] Capt. Grace Hopper on Future Possibilities: Data, Hardware, Software, and People (1982). https://www.nsa.gov, 2024. [Accessed 14-01-2025].

[24] European Police Chiefs call for industry and governments to take action against end-to-end encryption roll-out. www.europol.europa.eu, April 2024.

[25] Federal Rules of Civil Procedure. https://www.law.cornell.edu/rules/frcp, 2024. [Accessed 02-02-2025].

[26] Federal Rules of Criminal Procedure. https://www.law.cornell.edu/rules/frcrmp, 2024. [Accessed 02-02-2025].

[27] Federal Rules of Evidence. https://www.law.cornell.edu/rules/fre, 2024. [Accessed 02-02-2025].

[28] Wikipedia:Long-term abuse. https://en.wikipedia.org/wiki/Wikipedia:Long-term_abuse, 2025. [Accessed 20-01-2025].

[29] ABELSON, H., ANDERSON, R., BELLOVIN, S. M., BENALOH, J., BLAZE, M., DIFFIE, W., GILMORE, J., NEUMANN, P. G., RIVEST, R. L., SCHILLER, J. I., AND SCHNEIER, B. The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption. Center for Democracy and Technology report, May 1997. https://academiccommons.columbia.edu/doi/10.7916/D8R2176H/download.

[30] ABELSON, H., ANDERSON, R., BELLOVIN, S. M., BENALOH, J., BLAZE, M., DIFFIE, W. W., GILMORE, J., GREEN, M., LANDAU, S., NEUMANN, P. G., RIVEST, R. L., SCHILLER, J. I., SCHNEIER, B., SPECTER, M. A., AND WEITZNER, D. J. Keys under Doormats. *Commun. ACM 58*, 10 (sep 2015), 24–26.

[31] ADRIAN, D., BHARGAVAN, K., DURUMERIC, Z., GAUDRY, P., GREEN, M., HALDERMAN, J. A., HENINGER, N., SPRINGALL, D., THOMÉ, E., VALENTA, L., VANDERSLOOT, B., WUSTROW, E., ZANELLA-BÉGUELIN, S., AND ZIMMERMANN, P. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2015), CCS '15, Association for Computing Machinery, p. 5–17.

[32] ALHAQBANI, B., AND FIDGE, C. Privacy-preserving electronic health record linkage using pseudonym identifiers. In *HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services* (2008), pp. 108–117.

[33] BARTH, A. RFC 6265: HTTP State Management Mechanism. https://datatracker.ietf.org/doc/html/rfc6265, 2011. [Accessed 19-02-2025].

[34] BELLANGER, A. Silk Road founder Ross Ulbricht pardoned by Trump 10 years into life sentence. https://arstechnica.com/, 2025. [Accessed 01-02-2025].

[35] BENALOH, J., AND DE MARE, M. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In *Advances in Cryptology, EUROCRYPT '93* (Berlin, Heidelberg, 1994), T. Helleseth, Ed., Springer Berlin Heidelberg, pp. 274–285.

[36] BLAZE, M. Key escrow from a safe distance: looking back at the clipper chip. In *Proceedings of the 27th annual computer security applications conference* (2011), pp. 317–321.

[37] BOEYEN, S., SANTESSON, S., POLK, T., HOUSLEY, R., FARRELL, S., AND COOPER, D. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008.

[38] BONEH, D., ET AL. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS 46*, 2 (1999), 203–213.

[39] BROWN, J. The origin and early history of the office of notary. https://era.ed.ac.uk/handle/1842/29444, 1935. [Accessed 16-02-2025].

[40] CAMENISCH, J., AND LEHMANN, A. Privacy-preserving user-auditable pseudonym systems. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)* (2017), pp. 269–284.

[41] CARMICHAEL, R. D. Note on a new number theory function. *Bulletin of the American Mathematical Society 16*, 5 (1910), 232 – 238.

[42] CARMICHAEL, R. D. Fermat Numbers Fn = 22n + 1. *The American Mathematical Monthly 26*, 4 (1919), 137–146.

[43] CHAUM, D., AND VAN HEYST, E. Group Signatures. In *Advances in Cryptology, EUROCRYPT '91* (Berlin, Heidelberg, 1991), D. W. Davies, Ed., Springer Berlin Heidelberg, pp. 257–265.

[44] COHEN, C. *Biblical Hapax Legomena in the Light of Akkadian and Ugaritic*. Dissertation series. Scholars Press, 1978.

[45] DAY, J. D., AND ZIMMERMANN, H. The osi reference model. *Proceedings of the IEEE 71*, 12 (1983), 1334–1340.

[46] DEMUYNCK, L., AND DE DECKER, B. Privacy-preserving electronic health records. In *IFIP International Conference on Communications and Multimedia Security* (2005), Springer, pp. 150–159.

[47] DENNING, D. E. Digital signatures with RSA and other public-key cryptosystems. *Communications of the ACM 27*, 4 (1984), 388–392.

[48] DIFFIE, W., AND HELLMAN, M. New directions in cryptography. *IEEE Transactions on Information Theory 22*, 6 (1976), 644–654.

[49] DONG, K. Developing a Digital Property Law Regime. *Cornell Law Review 105*, 6 (2020). [Accessed 01-02-2025].

[50] DOUCEUR, J. R. The Sybil Attack. In *Peer-to-Peer Systems* (Berlin, Heidelberg, 2002), P. Druschel, F. Kaashoek, and A. Rowstron, Eds., Springer Berlin Heidelberg, pp. 251–260.

[51] DWORK, C., AND NAOR, M. An Efficient Existentially Unforgeable Signature Scheme and Its Applications. *Journal of Cryptology 11*, 3 (1998), 187–208.

[52] EDDY, W. M. RFC 9293: Transmission Control Protocol (TCP). `https://www.ietf.org/rfc/rfc9293.html`, 2022. [Accessed 08-02-2025].

[53] EPSTEIN, R. A. The Legal Regulation of Lawyers' Conflicts of Interest. *Fordham L. Rev. 60* (1991), 579.

[54] FAERBER, C. N. Being There: The Importance of Physical Presence to the Notary. *J. Marshall L. Rev. 31* (1997), 749.

[55] FIELDING, R. T., NOTTINGHAM, M., AND RESCHKE, J. HTTP Semantics. RFC 9110, June 2022.

[56] FINLEY, K. Web Commenters Claim They Use Pseudonyms for Privacy, Not Trolling. https://www.wired.com/2014/12/disqus/, 2014. [Accessed 20-01-2025].

[57] FISCHER, M. J., HOCHMAN, J. E., AND BOFFA, D. Privacy-Preserving Data Sharing for Medical Research. https://cpsc.yale.edu/sites/default/files/files/TR1558.pdf, 2021. [Accessed 16-02-2025].

[58] FISCHER, M. J., LYNCH, N. A., AND PATERSON, M. S. Impossibility of Distributed Consensus with One Faulty Process. *J. ACM 32*, 2 (Apr. 1985), 374–382.

[59] FRANKE, M., SEHILI, Z., AND RAHM, E. PRIMAT: A Toolbox for Fast Privacy-Preserving Matching. *Proc. VLDB Endow. 12*, 12 (Aug. 2019), 1826–1829.

[60] FREDERICK, B. Click Bots and Fake Traffic Cost Online Advertisers $35 Billion. https://www.searchenginejournal.com, 2022. [Accessed 20-01-2025].

[61] GILLIAN TAN, S. M. Startup ID.me to Let Shareholders Sell at $1.8 Billion Valuation. https://www.bnnbloomberg.com, 2024. [Accessed 17-02-2025].

[62] GONG, L. Variations on the themes of message freshness and replay-or the difficulty in devising formal methods to analyze cryptographic protocols. In *[1993] Proceedings Computer Security Foundations Workshop VI* (1993), pp. 131–136.

[63] GUERAR, M., VERDERAME, L., MIGLIARDI, M., PALMIERI, F., AND MERLO, A. Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-computer Dilemma. *ACM Comput. Surv. 54*, 9 (Oct. 2021).

[64] HELLMAN, M. An overview of public key cryptography. *IEEE Communications Society Magazine 16*, 6 (1978), 24–32.

[65] HOCHMAN, J., STEIN, J., AND CUNNINGHAM, J. US11570163B2 - User authentication system. https://patents.google.com. [Accessed 17-02-2025].

[66] KARIO, H. Everlasting ROBOT: the Marvin Attack. Cryptology ePrint Archive, Paper 2023/1442, 2023. https://eprint.iacr.org/2023/1442.

[67] KØIEN, G. M. A brief survey of nonces and nonce usage. *SECURWARE 2015* (2015), 97.

[68] LAMPORT, L. Time, Clocks, and the Ordering of Events in a Distributed System. *Commun. ACM 21*, 7 (July 1978), 558–565.

[69] LEACH, P., MEALLING, M., AND SALZ, R. RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace, 2005.

[70] LEVY, I., AND ROBINSON, C. Principles for a More Informed Exceptional Access Debate. https://www.lawfaremedia.org/article/principles-more-informed-exceptional-access-debate, 2018. [Accessed 15-03-2025].

[71] MATHISEN, H. N. Dancing with Fischer, Lynch & Paterson's Impossibility Result. Master's thesis, Reykjavík University, 2025.

[72] MATKOVIC, D. J. The Chinese Remainder Theorem: A Historical Account. *Pi Mu Epsilon Journal 8*, 8 (1988), 493–502.

[73] MCCLUSKEY, M. Inside the War on Fake Reviews. https://time.com/6192933/fake-reviews-regulation/, 2022. [Accessed 20-01-2025].

[74] MCENENEY, M. F., TEITELBAUM, D. E., AND KAUFMANN, K. F. Customer Identification Requirements Under the USA PATRIOT Act. *The Business Lawyer 59*, 3 (2004), 1287–1297.

[75] NEEDHAM, R. M., AND SCHROEDER, M. D. Using encryption for authentication in large networks of computers. *Commun. ACM 21*, 12 (dec 1978), 993–999.

[76] PARASKEVAS, A., AND BROOKES, M. Nodes, guardians and signs: Raising barriers to human trafficking in the tourism industry. *Tourism Management 67* (2018), 147–156.

[77] POHLIG, S., AND HELLMAN, M. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (Corresp.). *IEEE Transactions on Information Theory 24*, 1 (1978), 106–110.

[78] POMERANCE, C. A Tale of Two Sieves. *Notices of the AMS 43*, 12 (1996), 1473–1485.

[79] QUINE, W. V. O. *Word and object*. MIT press, 2013.

[80] REAGAN, R. T. *Sealing Court Records and Proceedings: A Pocket Guide*. DIANE Publishing, 2011.

[81] RICHARDSON, H. C. Letters from an American, January 26, 2025. https://heathercoxrichardson.substack.com/p/january-26-2025, 2025. [Accessed 30-01-2025].

[82] RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM 21*, 2 (Feb 1978), 120–126.

[83] RIVEST, R. L., SHAMIR, A., AND TAUMAN, Y. How to Leak a Secret. In *Advances in Cryptology, ASIACRYPT 2001* (Berlin, Heidelberg, 2001), C. Boyd, Ed., Springer Berlin Heidelberg, pp. 552–565.

[84] ROGERS, E. M. *Diffusion of innovations, Fifth Edition*. Free Press, Simon & Schuster, 2003.

[85] SHAMIR, A. How to share a secret. *Commun. ACM 22*, 11 (Nov. 1979), 612–613.

[86] SHOUP, V. *A Computational Introduction to Number Theory and Algebra*, 2nd ed. Cambridge University Press, Feb. 2009, ch. 5.5.5 Sophie Germain primes, pp. 123–124. ISBN: 9780521516440.

[87] SILVERMAN, C., AND KAO, J. Infamous Russian Troll Farm Appears to Be Source of Anti-Ukraine Propaganda. https://www.propublica.org, 2022. [Accessed 18-01-2025].

[88] SIMON, S. A lost hard drive in a landfill holds the key to a British man's crypto fortune. https://www.npr.org/, Feb 2025.

[89] STEEL, T. B. A first version of UNCOL. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference* (New York, NY, USA, 1961), IRE-AIEE-ACM '61 (Western), Association for Computing Machinery, p. 371–378.

[90] STEINER, J. G., NEUMAN, B. C., AND SCHILLER, J. I. Kerberos: An Authentication Service for Open Network Systems. In *Usenix Winter* (1988), pp. 191–202.

[91] TECHFUTURIST.TECH. DNS Failover and Redundancy for High Availability - Tech Futurist. `https://techfuturist.tech/dns-failover-and-redundancy-for-high-availability/`, 2024. [Accessed 15-01-2025].

[92] ULPH, J. The UCP 600: documentary credits in the twenty-first century. *The Journal of Business Law* (2007), 355–377.

[93] VANHOOKER, B. An Oral History of Whac-a-Mole. `https://melmagazine.com/en-us/story/whac-a-mole-oral-history`, 2020. [Accessed 15-03-2025].

[94] VATSALAN, D., CHRISTEN, P., AND VERYKIOS, V. S. A taxonomy of privacy-preserving record linkage techniques. *Information Systems 38*, 6 (2013), 946–969.