Unconditionally Secure Cryptography

Signature Schemes, User-Private Information Retrieval, and the Generalized Russian Cards Problem

by

Colleen M. Swanson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2013

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We focus on three different types of multi-party cryptographic protocols. The first is in the area of *unconditionally secure signature schemes*, the goal of which is to provide users the ability to electronically sign documents without the reliance on computational assumptions needed in traditional digital signatures. The second is on cooperative protocols in which users help each other maintain privacy while querying a database, called *user-private information retrieval protocols*. The third is concerned with the *generalized Russian cards problem*, in which two card players wish to communicate their hands to each other via public announcements without the third player learning the card deal. The latter two problems have close ties to the field of combinatorial designs, and properly fit within the field of *combinatorial cryptography*. All of these problems have a common thread, in that they are grounded in the *information-theoretically secure* or *unconditionally secure* setting.

Acknowledgements

I would like to thank my supervisor Douglas Stinson for his help and support throughout my thesis work. I wish to thank my thesis committee for their comments and suggestions.

Table of Contents

CHAPTER 1

Introduction

At the heart of cryptography is the desire to establish secure communication between one or more parties; cryptographic protocols (or systems) are the proffered solutions to specific instances of this problem. In general, cryptographic protocols may be categorized both according to the type of secrets, or *keying information*, the involved parties (or *users*) must possess, and according to the type of security the protocol affords, in particular the resources of the adversary the protocol claims to be secure against.

With respect to keying information, protocols may be grouped into two settings, *symmetric* and *public-key*. In symmetric cryptography (which is also called *secret-key cryptography*), communicating users rely on some shared secret or *secret key*; in this setting, the same key is used both to disguise and subsequently reveal the communicated information. In public-key cryptography (which is also called *asymmetric cryptography*), each user has a *key pair*, one part of which may be made public, termed a *public key*, and a second part, termed a *private key*, which should be known only to the user in question. In a typical example of the public-key setting, a user Bob disguises his message to Alice using her public key, and Alice then uses her private key to reveal the message.

Unsurprisingly, how we define the notion of a "secure" protocol depends heavily on the specifics of the given problem scenario, including the goals of the parties involved and our understanding of an *adversary*—his capabilities and goals—which we describe in a *threat* or *adversarial model*. Broadly speaking, there are two main approaches to security analysis, depending on the resources afforded the adversary, namely *computational* and *unconditional*.

A common approach is to consider the *computational security* of schemes, in which the adversary is assumed to have limited computational resources available, such as bounded memory and computing power. Informally, when we discuss security in this setting, we desire that relevant attacks be *computationally infeasible* for an adversary to launch, meaning that the adversary has only a small chance of success. In the realm of *provable security*, a subclass of computational security, we rest our security arguments on the assumption that a particular problem is suitably *hard* for an adversary to solve (given his bounded resources), and show that breaking the security of the given scheme amounts to solving the hard problem. Typical examples of hard problems include factoring large composite numbers and computing discrete logarithms in certain groups.

The adversarial approach taken in this thesis, however, is that of *unconditionally* or *information-theoretically secure cryptography*. In this setting, we assume the adversary has access to *unlimited* computational resources. In particular, we can no longer use the expected amount of work (or estimated time) needed to complete an attack as a measure of security. In this respect, public-key protocols properly fit within the computational security setting; as the public and private components of a key pair are used for complementary functions—to disguise and reveal—an adversary with unlimited resources can always launch a "brute force" attack on the system using knowledge gleaned from the public key. Hence, unconditionally secure protocols rely on either shared secrets, as in the symmetric setting, or require a distributed setting, in which users each have access to different *shares*, or portions, of the secret information used in the set-up phase.

Information-theoretically secure cryptography explores what types of protocols are possible if we remove the standard computational assumptions as to the physical capabilities of our adversaries. In particular, if we remove our reliance on the supposed hardness of various problems, what may be achieved? In this sense, information-theoretically secure cryptography fits well in the context of *post-quantum cryptography*—cryptography that remains secure in the advent of quantum computers—now a popular area of research. Whether or not one subscribes to the belief that a large quantum computer is on the horizon, or merely wishes to ensure the security of information in the long term (independent of any breakthroughs in solving difficult open problems), the study of unconditionally secure cryptography is certainly interesting from a theoretical perspective and has the potential to be of real-world importance.

Moreover, information-theoretically secure cryptography is a broad discipline that relies heavily on theoretical mathematics, drawing from such realms as information theory, probability theory, and combinatorics. As such, many of the results may be appreciated for the beauty of the underlying mathematics as well as for applicability to the real world.

In this thesis, we focus on three different types of multi-party cryptographic protocols. The first is in the area of *unconditionally secure signature schemes*, the goal of which is to provide users the ability to electronically sign documents without the reliance on computational assumptions needed in traditional digital signatures. The second is on cooperative protocols in which users help each other maintain privacy while querying a database, called *user-private information retrieval protocols*. The third is concerned with the *generalized Russian cards problem*, in which two card players wish to communicate their hands to each other via public announcements without the third player learning the card deal. The latter two problems have close ties to the field of combinatorial designs, and properly fit within the field of *combinatorial cryptography*. All of these problems are grounded in the *information-theoretically secure* or *unconditionally secure* setting.

In the remainder of this chapter, we begin with an introduction to unconditionally secure cryptography and define some basic cryptographic primitives that will prove useful. For clarity, the primitives we discuss are given in both the computational and unconditional

security contexts. We then provide the relevant background from the field of combinatorial designs.

1.1. Unconditionally Secure Cryptography

Our approach is motivated by the classic approach to secrecy introduced by Shannon [64]. In particular, the security analysis of all our protocols is phrased in terms of probability distributions on the information available to the various parties. As such, we feel it is useful to include a brief introduction to unconditional security. The concepts discussed in this section are standard in the field; much of the material and notation follows the presentation in Stinson [69] and Katz and Lindell [45].

In unconditionally secure cryptographic protocols, the adversary's advantage in breaking a given scheme should be limited to some prespecified probability. That is, the information available to an adversary $\mathcal{A}$ after the protocol is completed should not allow $\mathcal{A}$ to do more than guess at the secrets protected by the scheme, and the success probability of this guess is fixed no matter the computational resources of $\mathcal{A}$.

For example, suppose we have a *cryptosystem* with a finite set of possible *plaintexts* $\mathcal{P}$, a finite set of possible *ciphertexts* $\mathcal{C}$, a finite set of possible *keys* $\mathcal{K}$ and *encryption* and *decryption rule* sets $\mathcal{E} = \{e_K : \mathcal{P} \to \mathcal{C}\}_{K\in\mathcal{K}}$ and $\mathcal{D} = \{d_K : \mathcal{C} \to \mathcal{P}\}_{K\in\mathcal{K}}$, respectively. Each key $K \in \mathcal{K}$ corresponds to an encryption rule $e_K \in \mathcal{E}$ and decryption rule $d_K \in \mathcal{D}$ satisfying $d_K(e_K(x)) = x$ for every plaintext $x \in \mathcal{P}$. Here we consider cryptosystems in which a key $K \in \mathcal{K}$ is used only once. The goal of a cryptosystem is to allow a user Alice to send a message to another user Bob over an insecure channel, such that an eavesdropping adversary Eve cannot determine the message, but Bob can. Here we assume that Alice and Bob use a predetermined key $K \in \mathcal{K}$ that is kept secret from Eve. In particular, this is a *symmetric cryptosystem*. In analyzing the properties of such a scheme, we concern ourselves with Eve's ability, on seeing some ciphertext $c \in \mathcal{C}$, to determine a corresponding plaintext x.

Here probability theory comes into play. Eve has unlimited computational resources, so we must consider Eve's *a priori* knowledge of the scheme, in particular how likely individual plaintexts, keys, and ciphertexts are to occur, and her *a posteriori* knowledge, namely the observed ciphertext c. Eve can use this knowledge to determine the likelihood that c corresponds to a given plaintext message x. She can do this for all possible messages in $\mathcal{P}$ and the resulting set of probabilities represents the advantage Eve has in guessing the message corresponding to c. In particular, if the *a posteriori* probability that a plaintext is x (given Eve's knowledge of c) is the same as the *a priori* probability that the plaintext is x, then we have *perfect secrecy*, a concept introduced by Shannon [64].

Formally, we express these concepts in terms of *discrete random variables* and *probability distributions*. The following concepts from elementary probability theory are useful.

Definition 1.1. A *discrete random variable*, say $\mathbf{Y}$, is a finite set Y and a *probability distribution* on Y that associates each element $y \in Y$ with a probability. We use $\Pr[\mathbf{Y} = y]$ to denote the probability that the random variable $\mathbf{Y}$ has value y and require $\Pr[\mathbf{Y} = y] \geq 0$ for every $y \in Y$. The probability distribution satisfies $\sum_{y \in Y} \Pr[\mathbf{Y} = y] = 1$.

Let $\mathbf{X}$ and $\mathbf{Y}$ be probability distributions on finite sets X and Y, respectively.

Definition 1.2. The *joint probability distribution on* $\mathbf{X}$ *and* $\mathbf{Y}$, denoted $\Pr[\mathbf{X} = x, \mathbf{Y} = y]$, is the probability that $\mathbf{X}$ has value x and $\mathbf{Y}$ has value y. The *conditional probability distribution on* $\mathbf{X}$ *given* $\mathbf{Y}$, denoted $\Pr[\mathbf{X} = x \mid \mathbf{Y} = y]$ is the probability distribution of $\mathbf{X}$ when $\mathbf{Y}$ is known to be a particular value y. In particular, we have the following formula relating the joint and conditional probabilities:

$$\Pr[\mathbf{X} = x, \mathbf{Y} = y] = \Pr[\mathbf{X} = x \mid \mathbf{Y} = y] \times \Pr[\mathbf{Y} = y].$$

Definition 1.3. The random variables $\mathbf{X}$ and $\mathbf{Y}$ are said to be *independent* if

$$\Pr[\mathbf{X} = x, \mathbf{Y} = y] = \Pr[\mathbf{X} = x] \times \Pr[\mathbf{Y} = y]$$

for all $x \in X$, $y \in Y$.

A useful result relating conditional probability distributions is *Bayes' Theorem*:

Theorem 1.1 (Bayes' Theorem)**.** *If* $\Pr[\mathbf{Y} = y] > 0$*, then*

$$\Pr[\mathbf{X} = x \mid \mathbf{Y} = y] = \frac{\Pr[\mathbf{X} = x] \times \Pr[\mathbf{Y} = y \mid \mathbf{X} = x]}{\Pr[\mathbf{Y} = y]}.$$

In terms of the cryptosystem above, we define random variables $\mathbf{P}$, $\mathbf{K}$, and $\mathbf{C}$ denoting the set of plaintexts, keys, and ciphertexts, respectively. Here $\mathbf{P}$ and $\mathbf{K}$ are assumed to be independent random variables; together the associated probability distributions determine the probability distribution on $\mathbf{C}$. Eve's *a priori* knowledge of the scheme is represented by these probability distributions. Once Eve observes a ciphertext c, she can use the conditional probability distribution $\Pr[\mathbf{P} = x \mid \mathbf{C} = c]$ to determine the probability that c corresponds to a given message $x \in \mathcal{P}$. We have the following formal definition of perfect secrecy:

Definition 1.4. A cryptosystem satisfies *perfect secrecy* if

$$\Pr[\mathbf{P} = x \mid \mathbf{C} = c] = \Pr[\mathbf{P} = x]$$

for all $x \in \mathcal{P}$ and all $c \in \mathcal{C}$.

In fact, a cryptosystem satisfies perfect secrecy if and only if the random variables $\mathbf{P}$ and $\mathbf{C}$ are independent.

We can apply these notions to the security analysis of other types of cryptographic protocols, again by considering the information available to an adversary after a protocol

execution and comparing probability distributions. That is, we can use probability distributions to formally express the advantage of an adversary in guessing the relevant secret. The concept of perfect secrecy also applies: if the probability that the relevant secret takes on value s given the adversary's observations is the same as the *a priori* probability that the secret is s for all possible secrets s, then we have perfect secrecy.

Unsurprisingly, many of the security arguments in the realm of unconditionally secure cryptography have a similar flavor, taking on the form of analyzing probability distributions with respect to the information available to an adversary before and after a protocol execution. The proof that the *one-time pad* cryptosystem satisfies perfect secrecy is a classic example of this type of argument, so we include it here.

EXAMPLE 1.1 (One-time pad cryptosystem). Let $n \geq 1$ be an integer and let $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$. For K, $x \in (\mathbb{Z}_2)^n$, define both the encryption rule $e_K(x)$ and decryption rule $d_K(x)$ to be the sum of K and x modulo 2. Suppose every key K is used with equal probability.

In particular, this cryptosystem is perfectly secure. The key observation is that for each pair $(x, c) \in \mathcal{P} \times \mathcal{C}$, there is a unique key $K \in \mathcal{K}$ such that $e_K(x) = c$. Fix an arbitrary ciphertext $c \in (\mathbb{Z}_2)^n$. Then

$$\begin{aligned}
\Pr[\mathbf{C} = c] &= \sum_{K \in (\mathbb{Z}_2)^n} \Pr[\mathbf{K} = K] \times \Pr[\mathbf{P} = d_K(c)] \\
&= \frac{1}{2^n} \sum_{K \in (\mathbb{Z}_2)^n} \Pr[\mathbf{P} = d_K(c)] \\
&= \frac{1}{2^n} \sum_{x \in (\mathbb{Z}_2)^n} \Pr[\mathbf{P} = x] \\
&= \frac{1}{2^n}
\end{aligned}$$

and

$$\Pr[\mathbf{C} = c \mid \mathbf{P} = x] = \Pr[\mathbf{K} = (c - x) \bmod 2] = \frac{1}{2^n}.$$

This yields

$$\Pr[\mathbf{P} = x \mid \mathbf{C} = c] = \frac{\Pr[\mathbf{C} = c \mid \mathbf{P} = x] \times \Pr[\mathbf{P} = x]}{\Pr[\mathbf{C} = c]} = \Pr[\mathbf{P} = x]$$

for all $x \in \mathcal{P}$ and $c \in \mathcal{C}$, as desired.

1.1.1. Cryptographic primitives. In this section, we introduce the cryptographic primitives *message authentication codes* and *digital signatures*.

We first discuss a useful security concept, that of *negligible success probability*. In general, for a cryptographic primitive to be secure, we want the probability of a successful attack to be *negligible*, which is typically understood to mean that the success probability

of an attacker is smaller than any inverse polynomial in a security parameter k. That is, we formalize the notion of negligible as follows.

Definition 1.5. Let f be a non-negative function of k. If for every positive polynomial $p(\cdot)$, there exists an N such that for all integers $k > N$, it holds that $f(k) < \frac{1}{p(k)}$, then we say that the function f is *negligible*.

A message authentication code is a special type of *keyed hash family*; we introduce these concepts with the computational setting in mind and then move to the unconditionally secure setting.

Definition 1.6. A *keyed hash family* is a tuple $(X, Y, \mathcal{K}, \mathcal{H})$, where the following hold:

1. X is a set of possible messages.
2. Y is a finite set of possible *authentication tags*.
3. $\mathcal{K}$, the *keyspace*, is a finite set of possible *keys*.
4. For each $K \in \mathcal{K}$, there is a *hash function* $h_K \in \mathcal{H}$, where $h_K \colon X \to Y$.

Definition 1.7. A *message authentication code (MAC)* is a keyed hash family $(X, Y, \mathcal{K}, \mathcal{H})$ that satisfies the following property, known as *computation-resistance*:

- Given zero or more pairs of the form $(x_i, h_K(x_i)) \in X \times Y$ and assuming K is not known, it is computationally infeasible to compute any pair $(x, h_K(x))$ for any $x \neq x_i$ for all i. That is, an attacker without knowledge of K should have a negligible probability of success in creating such a pair.

A message authentication code $(X, Y, \mathcal{K}, \mathcal{H})$ is typically used to allow a sender, Alice, to send a message over an insecure channel to a receiver, Bob. In particular, Alice and Bob share a secret $K \in \mathcal{K}$, and Alice sends her message, say $x \in X$, together with the corresponding authentication tag, $h_K(x)$, to Bob. Upon receipt, Bob is sure both that the message has not been altered and that Alice is the person who actually sent the message. As Alice and Bob have a shared secret key, MACs are an example of a symmetric or secret-key cryptographic primitive.

If an adversary $\mathcal{A}$ can produce a message x and corresponding authentication tag $h_K(x)$ for some previously unseen message x, i.e., if the MAC does not satisfy computation-resistance, we call the pair $(x, h_K(x))$ a *MAC forgery*. In creating such a forgery, we can specify the amount of control the adversary $\mathcal{A}$ has over the pairs $(x_i, h_K(x_i))$ to which he has access. If $\mathcal{A}$ can successively pick the messages x_i after seeing the corresponding authentication tags for his previous selections, but has no control over x, then forgery $(x, h_K(x))$ is an *existential forgery under adaptive chosen-message attack*. If the adversary has (at least some) control over x, then $(x, h_K(x))$ is a *selective forgery under adaptive chosen-message attack*. Both of these are typical attack scenarios on protocols in traditional public-key cryptography.

In the unconditionally secure setting, a key is typically used to produce just one authentication tag. In attempting to produce a MAC forgery $(x, h_K(x))$, the adversary typically has access to either one valid pair $(x_1, h_K(x_1))$ (a *substitution attack*), or access to no valid pairs (an *impersonation attack*).

We observe that since Alice and Bob both know the secret K, either party can produce a valid pair $(x, h_K(x))$. Alice can deny having sent a given message and corresponding authentication tag by claiming that Bob could just as easily have created the pair. That is, MACs cannot be used in situations where the ability to *repudiate* sending a message is undesirable. In addition, the purpose of a MAC is to establish message integrity and sender identity *between Alice and Bob*, not to convince outside parties. Only those who know the secret K can determine whether a given message/authentication tag pair is valid under K and only Alice and Bob should know K.

These properties are important when we consider *digital signatures*, which are meant to emulate traditional pen-and-paper signatures in the digital world. In the paper world, we have the following desirable properties: we want signatures to be unique and difficult for others to reproduce or *forge*, we want to be confident that no one can sign a document and then later deny or *repudiate* his signature, and we want to be confident that if we accept someone's signature as being real or *valid*, the signature is *transferable* in the sense that other people will also accept this signature as valid.

Typically, a digital signature involves a *private signing algorithm*, which a signer uses to sign messages, and a *public verification algorithm*, with which anyone can verify the validity of a signature on a particular message. That is, digital signatures are an example of cryptographic primitives in the asymmetric or public-key setting. Formally, we can define a *signature scheme* in traditional public-key cryptography in the following manner.

Definition 1.8. A *signature scheme* is a tuple $(X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfying the following:

1. X is a finite set of possible messages.
2. Σ is a finite set of possible signatures.
3. The *key-generation algorithm* Gen takes as input 1^k, where k is a security parameter, and outputs a pair of keys (pk, sk), where pk is the *public key* and sk is the *private key*.
4. The *signing algorithm* Sign takes a private key sk and a message $x \in \mathcal{X}$ as input, and outputs a signature $\sigma \in \Sigma$, denoted as $\mathsf{Sign}_{sk}(x)$.
5. The deterministic *verification algorithm* Vrfy takes a message $x \in \mathcal{X}$, a signature $\sigma \in \Sigma$, and a public key pk as input, and outputs either *valid* or *invalid*, denoted as $\mathsf{Vrfy}_{pk}(x, \sigma)$.

It is required that, for every k, for every pair (pk, sk) output by $\mathsf{Gen}(1^k)$, and for every $x \in \mathcal{X}$, it holds that

$$\mathsf{Vrfy}_{pk}\left(x, \mathsf{Sign}_{sk}(x)\right) = \mathit{valid}.$$

Furthermore, the algorithms Gen, Sign, and Vrfy should be polynomial-time algorithms.

REMARK 1.1. The signing algorithm in Definition 1.8 may be deterministic or randomized.

We say a signature pair (x, σ) is *valid* if $\mathsf{Vrfy}_{pk}(x, \sigma) = \mathit{valid}$. Stated informally, the security properties a signature scheme should satisfy are listed below.

1. *Unforgeability*: Except with negligible probability (with respect to the given security parameter k), it should not be possible to create a valid signature without knowledge of the corresponding private key. This corresponds to the real-world notion that each person should have a unique signature that is difficult for others to reproduce.
2. *Non-repudiation*: Except with negligible probability (with respect to the given security parameter k), a signer should be unable to repudiate a legitimate signature that he has created.
3. *Transferability*: Except with negligible probability (with respect to the given security parameter k), if a verifier accepts a signature, he can be confident that any other verifier will also accept it.

We can formally model attacks on signature schemes by a game in which the adversary is given access to a collection of valid signature pairs via a *signing oracle*. Here, a signing oracle can be thought of as a "black box" that outputs valid signature pairs. The adversary wins the game if he successfully outputs a valid signature pair on some message x for which he has not already seen a corresponding signature. We give a formal definition for this game and a corresponding definition of security as follows.

Definition 1.9. Let $\Pi = (X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with security parameter k and let $\mathcal{A}$ be a (polynomial-time) adversary. We define the following *signature game* $\text{Sig-forge}_{\mathcal{A},\Pi}(k)$:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair (pk, sk).
2. The adversary $\mathcal{A}$ is given pk and oracle access to Sign_{sk}. This oracle, which we denote by $\mathsf{Sign}^{\mathcal{O}}_{sk}$, takes as input a message $x' \in X$ of $\mathcal{A}$'s choice and outputs a signature $\mathsf{Sign}_{sk}(x') \in \Sigma$. We let $\mathcal{Q}$ denote the set of messages that the adversary $\mathcal{A}$ submitted as queries to the oracle $\mathsf{Sign}^{\mathcal{O}}_{sk}$.
3. The adversary $\mathcal{A}$ outputs a signature pair (x, σ).
4. The output of the game is defined to be 1 if and only if the following two conditions are met:
   (a) $x \notin \mathcal{Q}$ and
   (b) $\mathsf{Vrfy}_{pk}(x, \sigma) = \mathit{valid}$.

Definition 1.10. Let $\Pi = (X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme with security parameter k. We say Π is *existentially unforgeable under adaptive chosen-message attack* if for all polynomial-time adversaries $\mathcal{A}$, there exists a negligible function ϵ such that

$$\Pr\left[\text{Sig-forge}_{\mathcal{A},\Pi}(k) = 1\right] \leq \epsilon(k).$$

We make a few final remarks on signature schemes in the traditional public-key setting, in order to set the stage for our study of *unconditionally secure signatures* in Chapter 2. So long as the signature scheme in use is unforgeable (i.e., as in Definition 1.10), then transferability and non-repudiation are also satisfied. A direct consequence of the public nature of the verification algorithm and the unforgeability property of the signature scheme is that it is easy to tell whether or not a given signature is valid. A signer who attempts to repudiate a given valid signature will not be believed, as forgeries cannot realistically be produced. Moreover, once a person has verified the validity of a given signature pair, he can be confident that everyone else will agree as to the validity of the signature, since the same check will be used. As we will see in Chapter 2, the validity of signatures (and hence the properties of unforgeability, non-repudiation, and transferability) is not as obvious in the unconditionally secure setting, in which the adversary has unlimited computational resources.

1.2. Combinatorial Designs

In this section, we present fundamental definitions and results from the theory of combinatorial designs, which are used in Chapter 3 and Chapter 4. In particular, we present only results that are needed in later chapters. For general references on this material, we refer the reader to Stinson [**68**] and Colbourn and Dinitz [**11**]. We use notation and results from Stinson [**68**] throughout. Proofs of standard results are omitted and unless otherwise noted can be found in Stinson [**68**] or Colbourn and Dinitz [**11**].

Definition 1.11. A *set system* or *design* is a pair $(X, \mathcal{B})$ such that the following are satisfied:

1. X is a set of elements called *points*, and
2. $\mathcal{B}$ is a collection (i.e., a multiset) of nonempty proper subsets of X called *blocks*.

In the rest of this section, we abuse notation by writing blocks in the form abc instead of $\{a, b, c\}$. In addition, we make use of the following notation: for a positive integer t and a set X of size v, we let $\binom{X}{t}$ denote the set of $\binom{v}{t}$ t-subsets of X.

Definition 1.12. The *degree* of a point $x \in X$ is the number of blocks containing x. If all points have the same degree, r, we say $(X, \mathcal{B})$ is *regular (of degree r)*.

Definition 1.13. The *rank* of $(X, \mathcal{B})$ is the size of the largest block. If all blocks contain the same number of points, say k, then $(X, \mathcal{B})$ is *uniform (of rank k)*.

Definition 1.14. Let $(X, \mathcal{B})$ be a set system. The *dual* of $(X, \mathcal{B})$ is the set system $(\mathcal{B}, X)$, where $y \in \mathcal{B}$ is contained in $x \in X$ if and only if x is contained in y in $(X, \mathcal{B})$.

Definition 1.15. A *covering design* is a set system in which every pair of points occurs in at least one block.

EXAMPLE 1.2. A covering design.

$$X = \{1, 2, 3, 4, 5, 6, 7\} \text{ and } \mathcal{B} = \{13, 23, 157, 124, 347, 356, 2567, 14567\}.$$

Definition 1.16. A *pairwise balanced design* (or *PBD*) is a set system such that every pair of distinct points is contained in exactly λ blocks, where λ is a fixed positive integer. Note that any PBD is a covering design.

EXAMPLE 1.3. A PBD with $\lambda = 2$.

$$X = \{1, 2, 3, 4, 5\} \text{ and } \mathcal{B} = \{12, 25, 135, 145, 1234, 2345\}.$$

Definition 1.17. Let $(X, \mathcal{B})$ be a regular and uniform set system of degree r and rank k, where $|X| = v$ and $|\mathcal{B}| = b$. Then we say $(X, \mathcal{B})$ is a (v, b, r, k)-1-*design*.

REMARK 1.2. The notation for a (v, b, r, k)-1-design $(X, \mathcal{B})$ reflects the fact that each point occurs in a constant number of blocks. Such a design $(X, \mathcal{B})$, sometimes referred to simply as a 1-*design*, is a special case of a t-design, which we introduce for $t \geq 2$ in Definition 1.24.

EXAMPLE 1.4. A $(5, 5, 3, 3)$-1-design.

$$X = \{1, 2, 3, 4, 5\} \text{ and } \mathcal{B} = \{123, 451, 234, 512, 345\}.$$

1.2.1. Balanced Incomplete Block Designs and Configurations.

Definition 1.18. Let v, k, and λ be positive integers such that $v > k \geq 2$. A (v, k, λ)-*balanced incomplete block design* (or (v, k, λ)-*BIBD*) is a set system $(X, \mathcal{B})$ such that the following are satisfied:

1. $|X| = v$,
2. each block contains exactly k points, and
3. every pair of distinct points is contained in exactly λ blocks.

Theorem 1.2. *Let $(X, \mathcal{B})$ be a (v, b, r, k, λ)-BIBD. Then every point has degree*

$$r = \frac{\lambda(v-1)}{k-1}$$

and the number of blocks is precisely

$$b = \frac{vr}{k} = \frac{\lambda(v^2 - v)}{k^2 - k}.$$

REMARK 1.3. We sometimes make all five parameters explicit by writing (v, b, r, k, λ)-BIBD instead of (v, k, λ)-BIBD.

REMARK 1.4. A (v, b, r, k, λ)-BIBD can be viewed as a (v, b, r, k)-1-design in which every pair of points occurs in exactly λ blocks. Equivalently, a (v, b, r, k, λ)-BIBD is a PBD that is regular and uniform of degree r and rank k.

Theorem 1.3. (Fisher's Inequality) *In any (v, b, r, k, λ)-BIBD, $b \geq v$.*

EXAMPLE 1.5. A $(10, 15, 6, 4, 2)$-BIBD.

$$X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

and

$$\mathcal{B} = \{0123, 0147, 0246, 0358, 0579, 0689, 1258,$$
$$1369, 1459, 1678, 2379, 2489, 2567, 3478, 3456\}.$$

Definition 1.19. A (v, k, λ)-BIBD is *simple* if every block occurs with multiplicity one.

Definition 1.20. A (v, k, λ)-BIBD in which every pair of blocks intersect in at most two points is *supersimple*.

Definition 1.21. A *symmetric BIBD* is a BIBD in which $b = v$.

Theorem 1.4. *In a symmetric BIBD, any two blocks intersect in exactly λ points.*

Definition 1.22. A *Steiner triple system of order v*, denoted STS(v), is a $(v, 3, 1)$-BIBD.

REMARK 1.5. It is well known that an STS(v) exists if and only if $v \equiv 1, 3 \bmod 6$, $v \geq 7$.

EXAMPLE 1.6. An STS(7).

$$X = \{0, 1, 2, 3, 4, 5, 6\} \text{ and } \mathcal{B} = \{013, 124, 235, 346, 045, 156, 026\}.$$

Definition 1.23. A (v, b, r, k)-*configuration* is a (v, b, r, k)-1-design such that every pair of distinct points is contained in at most 1 block.

EXAMPLE 1.7. A $(9, 9, 3, 3)$-configuration.

$$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \text{ and } \mathcal{B} = \{147, 258, 369, 159, 267, 348, 168, 249, 357\}.$$

REMARK 1.6. A (v, b, r, k)-configuration with $v = r(k-1)+1$ is a $(v, b, r, k, 1)$-BIBD.

1.2.2. t-designs.

Definition 1.24. Let v, k, λ, and t be positive integers with $v > k \geq t$. A t-(v, k, λ)-*design* is a set system $(X, \mathcal{B})$ such that the following are satisfied:

1. $|X| = v$,
2. each block contains exactly k points, and
3. every subset of t distinct points from X occurs in precisely λ blocks.

Definition 1.25. A t-(v, k, λ)-design $(X, \mathcal{B})$ is *simple* if every block in $\mathcal{B}$ occurs with multiplicity one.

REMARK 1.7. A 2-(v, k, λ)-design is just a (v, k, λ)-BIBD.

Definition 1.26. The design formed by taking λ copies of every k-subset of a v-set as blocks is a t-$\left(v, k, \lambda\binom{v-t}{k-t}\right)$-design, called a *trivial t-design*.

The following theorems are standard results for t-designs:

Theorem 1.5. *Let $(X, \mathcal{B})$ be a t-(v, k, λ)-design. Let $Z \subseteq X$ such that $|Z| = i < t$. Then $(X \backslash Z, \{B \backslash Z : Z \subseteq B \in \mathcal{B}\})$ is a $(t-i)$-$(v-i, k-i, \lambda)$-design.*

Theorem 1.6. *Let $(X, \mathcal{B})$ be a t-(v, k, λ)-design. Let $Y \subseteq X$ such that $|Y| = s \leq t$. Then there are precisely*

$$\lambda_s = \frac{\lambda \binom{v-s}{t-s}}{\binom{k-s}{t-s}}$$

blocks in $\mathcal{B}$ that contain Y.

Corollary 1.7. *Let $(X, \mathcal{B})$ be a t-(v, k, λ)-design and suppose $1 \leq s \leq t$. Then $(X, \mathcal{B})$ is an s-(v, k, λ_s)-design, where*

$$\lambda_s = \frac{\lambda \binom{v-s}{t-s}}{\binom{k-s}{t-s}}.$$

Theorem 1.8. *Let $(X, \mathcal{B})$ be a t-(v, k, λ)-design. Let $Y \subseteq X$ and $Z \subseteq X$ such that $Y \cap Z = \emptyset$, $|Y| = i$, $|Z| = j$, and $i + j \leq t$. Then there are precisely*

$$\lambda_i^j = \frac{\lambda \binom{v-i-j}{k-i}}{\binom{v-t}{k-t}}$$

blocks in $\mathcal{B}$ that contain all the points in Y and none of the points in Z.

Example 1.8. A 3-$(8, 4, 1)$-design.

$$X = \{0, 1, 2, 3, 4, 5, 6, 7\} \text{ and}$$

$$\mathcal{B} = \{3456, 2567, 2347, 1457, 1367, 1246, 1235, 0467, 0357, 0245, 0236, 0156, 0134, 0127\}.$$

Definition 1.27. A t-$(v, k, 1)$-design is called a *Steiner system with parameters t, k, v* and is denoted by $\mathrm{S}(t, k, v)$.

Remark 1.8. A Steiner triple system of order v, or $\mathrm{STS}(v)$, is an $\mathrm{S}(2, 3, v)$, i.e., a Steiner system in which $k = 3$.

Definition 1.28. A *large set of t-$(v, k, 1)$-designs* is a set $\{(X, \mathcal{B}_1), \ldots, (X, \mathcal{B}_N)\}$ of t-$(v, k, 1)$-designs (all of which have the same point set, X), in which every k-subset of X occurs as a block in precisely one of the $\mathcal{B}_i$s. That is, the $\mathcal{B}_i$s form a partition of $\binom{X}{k}$.

Remark 1.9. It is easy to prove that there must be exactly $N = \binom{v-t}{k-t}$ designs in a large set of t-$(v, k, 1)$-designs.

Remark 1.10. There are $v - 2$ designs in a large set of $\mathrm{STS}(v)$. It is known that a large set of $\mathrm{STS}(v)$ exists if and only if $v \equiv 1, 3 \bmod 6$ and $v \geq 9$.

Example 1.9. A large set of STS(9) [49].

$$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \text{ and } \mathcal{B}_1, \ldots, \mathcal{B}_7,$$

where the 7 block sets $\mathcal{B}_1, \ldots, \mathcal{B}_7$ are given by the rows of the following table:

123	145	169	178	249	257	268	348	356	379	467	589
124	136	158	179	235	267	289	349	378	457	468	569
125	137	149	168	238	247	269	346	359	458	567	789
126	139	148	157	234	259	278	358	367	456	479	689
127	135	146	189	239	248	256	347	368	459	578	679
128	134	159	167	236	245	279	357	389	469	478	568
129	138	147	156	237	246	258	345	369	489	579	678

1.2.3. Some Constructions of t-designs. We now discuss some existence results and constructions for t-designs. In general, constructing t-designs is a difficult problem. In the following, we give a brief overview of each construction; details may be found in either Stinson [68] or Colbourn and Dinitz [11], unless otherwise noted.

1.2.3.1. *Projective planes.*

Definition 1.29. A $(q^2+q+1, q+1, 1)$-BIBD is called a *finite projective plane of order q.*

Remark 1.11. Projective planes are symmetric BIBDs.

Suppose q is a prime power. Projective planes of order q can be constructed by considering the one-dimensional and two-dimensional subspaces of $(\mathbb{F}_q)^3$. In particular, each two-dimensional subspace A gives rise to a block B consisting of all one-dimensional subspaces contained in A. This construction yields the following result:

Theorem 1.9. *A finite projective plane of order q exists for every prime power $q \geq 2$.*

This construction method generalizes to higher dimensions, yielding symmetric BIBDs whose blocks are *hyperplanes* in the associated geometry.

Theorem 1.10. *There exists a symmetric $\left(\frac{q^{d+1}-1}{q-1}, \frac{q^d-1}{q-1}, \frac{q^{d-1}-1}{q-1}\right)$-BIBD for every integer $d \geq 2$ and prime power $q \geq 2$.*

1.2.3.2. *Large sets of Steiner triple systems.* There is a nice construction of large sets of Steiner triple systems of order v for certain choices of v. This construction is due to Schreiber [61] and it is also presented by Wilson [81]; our presentation follows that of Wilson [81]. We need v to satisfy the following condition:

- If p is a prime divisor of $(v-2)$, then the order of (-2) modulo p is congruent to 2 modulo 4.

Let G be an abelian group (written additively) of order $v-2$, where $v-2$ satisfies the condition given above. In particular, $v-2$ is not divisible by 2 or 3.

Let $\mathcal{A}$ be the collection of all triples $A = \{x, y, z\}$ where $x+y+z = 0$ and x, y, $z \in G$ are distinct. Note that a pair $\{x, y\}$ is contained in some 3-subset A of $\mathcal{A}$ exactly when

the solution z to $x+y+z=0$ is not equal to x or y. In particular, the zero element of G occurs with every other nonzero element of G in precisely one 3-subset $A \in \mathcal{A}$, since 2 does not divide the order of G. Consider the set of pairs $\{x, y\}$ which are not contained in some 3-subset A of $\mathcal{A}$; we call these *uncovered pairs*. The pair $\{x, y\}$ is uncovered precisely when $2x + y = 0$ or $x + 2y = 0$. In other words, pairs of the form $\{(-2)^{-1}x, x\}$ and $\{x, -2x\}$ for $x \neq 0 \in G$ do not appear in any 3-subset of $\mathcal{A}$.

Let Γ be the graph whose vertices are the nonzero elements of G and whose edge set is determined by the uncovered pairs. In particular, Γ is the union of disjoint cycles, where the length of the cycle containing a given $x \in G$ is the smallest positive integer ℓ such that $(-2)^\ell x = x$ in G. That is, the length of the cycle containing x is the (multiplicative) order of (-2) modulo m, where m is the (additive) order of x in G. Writing the prime factorization of m as $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$, the order of (-2) modulo m is the least common multiple of the orders of (-2) modulo $p_i^{\alpha_i}$ for $1 \leq i \leq r$. The last observation we need is that the order of (-2) modulo $p_i^{\alpha_i}$ is a nonnegative power of p_i times the order of (-2) modulo p_i. This is precisely what we need to see that these cycles have length congruent to 2 modulo 4, given the condition on the prime divisors of $(v-2)$.

Now, since Γ consists of cycles of even length, we can partition the edge set into two disjoint perfect matchings, say M_1 and M_2. We proceed by coloring all edges in M_1 blue and all edges in M_2 red.

We are now ready to construct an STS(v), which we denote by $S(0) = (X, \mathcal{B})$, on the point set $X = \{\infty_1, \infty_2\} \cup G$. Let $\mathcal{B}$ consist of the following blocks:

- $B_0 = \{\infty_1, \infty_2, 0\}$,
- each 3-subset $A \in \mathcal{A}$,
- for every blue edge $xy \in M_1$, we have a block $B_{xy} = \{\infty_1, x, y\}$, and
- for every red edge $xy \in M_2$, we have a block $B_{xy} = \{\infty_2, x, y\}$.

We then let G act on $S(0)$ additively to obtain the other $v-3$ Steiner triple systems. That is, for each element $g \in G$, we use a permutation on X that fixes the points ∞_1 and ∞_2 and for every $x \in G$ maps $x \mapsto x + g$. This yields a Steiner triple system of order v, which we denote by $S(g)$.

The fact that these are Steiner triple systems of order v is easy to check. The fact that these triple systems partition $\binom{X}{3}$ is a consequence of our divisibility assumptions on $(v-2)$. In brief, subsets of the form $\{\infty_1, \infty_2, g\}$ occur only in $S(g)$ for $g \in G$. Recalling that 3 is not a divisor of the order of G, we have that subsets of distinct triples $\{x, y, z\}$ of elements of G occur in $S(g)$ exactly when $x + y + z = 3g$. The crucial fact that subsets of the form $\{\infty_1, x, y\}$ and $\{\infty_2, x, y\}$ occur in exactly one of the $S(g)$ is equivalent to the fact that the length of each cycle in Γ is congruent to 2 modulo 4; for details, we refer the reader to Wilson [**81**].

EXAMPLE 1.10. Let $X = \mathbb{Z}_7 \cup \{\infty_1, \infty_2\}$. In this case the construction method above yields a graph Γ which is a single cycle of length 6. We can then use Γ to pick matchings

$M_1 = \{13, 45, 26\}$ and $M_2 = \{15, 46, 23\}$. This yields the following large set of STS(9):

$S(0)$:	$\infty_1\infty_20$	016	025	034	124	356	∞_115	∞_123	∞_146	∞_213	∞_226	∞_245
$S(1)$:	$\infty_1\infty_21$	120	136	145	235	460	∞_126	∞_134	∞_150	∞_224	∞_230	∞_256
$S(2)$:	$\infty_1\infty_22$	231	240	256	346	501	∞_130	∞_145	∞_161	∞_235	∞_241	∞_260
$S(3)$:	$\infty_1\infty_23$	342	351	360	450	612	∞_141	∞_156	∞_102	∞_246	∞_252	∞_201
$S(4)$:	$\infty_1\infty_24$	453	462	401	561	023	∞_152	∞_160	∞_113	∞_250	∞_263	∞_212
$S(5)$:	$\infty_1\infty_25$	564	503	512	602	134	∞_163	∞_101	∞_124	∞_261	∞_204	∞_223
$S(6)$:	$\infty_1\infty_26$	605	614	623	013	245	∞_104	∞_112	∞_135	∞_202	∞_215	∞_234

1.2.3.3. *Inversive planes.* We show there exists a 3-$(q^2+1, q+1, 1)$-design for all prime powers q by constructing an *inversive plane*; we briefly sketch this construction here. Let q be a prime power. Recall that the group $\mathrm{GL}(2, q)$ is the *general linear group* of all invertible matrices of dimension 2 over $\mathbb{F}_q$. The center of $\mathrm{GL}(2, q)$ is the group of all scalar matrices of dimension 2 over $\mathbb{F}_q$. The *projective linear group* $\mathrm{PGL}(2, \mathrm{q})$ is the quotient of $\mathrm{GL}(2, q)$ by its center.

Let $X = \mathbb{F}_{q^2} \cup \{\infty\}$ and $Y = \mathbb{F}_q \cup \{\infty\}$. Then $(X, \mathrm{orbit}(Y))$ is a 3-$(q^2+1, q+1, 1)$-design, namely, an *inversive plane*, where $\mathrm{orbit}(Y)$ is the orbit of subsets obtained by letting $\mathrm{PGL}(2, \mathrm{q}^2)$ act on Y. That this construction is a 3-design follows from the fact that $\mathrm{PGL}(2, q^2)$ is a sharply 3-transitive group acting on X; this means that for any choice of distinct x_1, x_2, $x_3 \in X$ and distinct y_1, y_2, $y_3 \in X$, there is exactly one element g in $\mathrm{PGL}(2, q^2)$ such that for $1 \leq i \leq 3$, the result of g acting on x_i is y_i.

In particular, this construction yields an infinite family of 3-designs. We have the following theorem:

Theorem 1.11. *For all prime powers q, there exists a 3-$(q^2+1, q+1, 1)$-design.*

1.2.3.4. *Hadamard designs.*

Definition 1.30. A *Hadamard design* is a symmetric $(4n-1, 2n-1, n-1)$-BIBD.

Hadamard designs are conjectured to exist for every $n \geq 2$. We present a construction for an infinite family of Hadamard designs based on *quadratic residue difference sets* in $\mathbb{F}_q$ for a prime power q satisfying $q \equiv 3 \bmod 4$. We need the following definitions:

Definition 1.31. Suppose G is a finite group of order v, written additively. Let k and λ be positive integers such that $v > k \geq 2$. A (v, k, λ)-*difference set in* G is a subset $D \subseteq G$ that satisfies the following conditions:

1. $|D| = k$,
2. the multiset $\{x - y : x, y \in D, x \neq y\}$ contains every nonzero element of G exactly λ times.

Definition 1.32. Let D be a (v, k, λ)-difference set in a group G. For any $g \in G$, the set $g + D$ is called a *translate* of D. The collection of all translates of D is called the *development of* D, denoted $\mathrm{Dev}(D)$.

The development of a difference set D in an abelian group G can be used to construct a symmetric BIBD. The following result is standard:

Theorem 1.12. *Let D be a (v, k, λ)-difference set in a group G. Then $(G, \mathrm{Dev}(D))$ is a symmetric (v, k, λ)-BIBD.*

Let q be an odd prime power and let $\mathrm{QR}(q) = \{x^2 : x \in \mathbb{F}_q, x \neq 0\}$ be the set of *quadratic residues* of $\mathbb{F}_q$. Then in particular, $\mathrm{QR}(q)$ is a difference set in $\mathbb{F}_q$ with the following parameters whenever $q \equiv 3 \bmod 4$:

Theorem 1.13. *Let q be a prime power such that $q \equiv 3 \bmod 4$. Then $\mathrm{QR}(q)$ is a $(q, \frac{q-1}{2}, \frac{q-3}{4})$-difference set in the additive group $\mathbb{F}_q$.*

This yields the following infinite family of Hadamard designs:

Corollary 1.14. *Let q be an odd prime power such that $q \equiv 3 \bmod 4$. Let $X = \mathbb{F}_q$ and $\mathcal{B} = \mathrm{Dev}(\mathrm{QR}(q))$. Then $(X, \mathcal{B})$ is a symmetric $(q, \frac{q-1}{2}, \frac{q-3}{4})$-BIBD.*

EXAMPLE 1.11. Let $q = 11$. Then $\mathrm{QR}(q) = \{1, 3, 4, 5, 9\}$ is an $(11, 5, 2)$-difference set D. Taking the development of D yields an $(11, 5, 2)$-BIBD $(X, \mathcal{B})$, where $X = \{0, \ldots, 9, \infty\}$ and $\mathcal{B}$ is the following collection of blocks:

$$\{13459, 2456\infty, 03567, 14678, 25789, 3689\infty, 0479\infty, 0158\infty, 01269, 1237\infty, 02348\}.$$

1.2.4. Transversal Designs.

Definition 1.33. Let t, v, k, and λ be positive integers satisfying $k \geq t \geq 2$. A *transversal design* $\mathrm{TD}_\lambda(t, k, v)$ is a triple $(X, \mathcal{G}, \mathcal{B})$ such that the following properties are satisfied:

1. X is a set of kv elements called *points*,
2. $\mathcal{G}$ is a partition of X into k subsets of size v called *groups*,
3. $\mathcal{B}$ is a set of k-subsets of X called *blocks*,
4. any group and any block contain exactly one common point, and
5. every subset of t points from distinct groups occurs in precisely λ blocks.

Definition 1.34. A $\mathrm{TD}_\lambda(t, k, v)$ is *simple* if there are no repeated blocks.

Many of the standard results for t-designs can be extended to transversal designs. The following terminology and results are useful:

Definition 1.35. Let $(X, \mathcal{G}, \mathcal{B})$ be a $\mathrm{TD}_\lambda(t, k, v)$ and write $\mathcal{G} = \{G_j : 1 \leq j \leq k\}$. Suppose $Z \subseteq X$ such that $|Z| = i \leq k$ and $|Z \cap G_j| \leq 1$ for $1 \leq j \leq k$. We say Z is a *partial transversal of* $\mathcal{G}$. If $i = k$, then we say Z is a *transversal of* $\mathcal{G}$.

Definition 1.36. For a partial transversal Z of $\mathcal{G}$, we let $G_Z = \{G_j \in \mathcal{G} : Z \cap G_j \neq \emptyset\}$ denote the set of groups that intersect Z. If Y, $Z \subseteq X$ are partial transversals of $\mathcal{G}$ such that $G_Z \cap G_Y = \emptyset$, we say Y, Z are *group disjoint*.

Theorem 1.15. *Let $(X, \mathcal{G}, \mathcal{B})$ be a $\mathrm{TD}_\lambda(t, k, v)$ and write $\mathcal{G} = \{G_j : 1 \leq j \leq k\}$. Suppose $Z \subseteq X$ is a partial transversal of $\mathcal{G}$ such that $|Z| = i < t$. Let*

$$\mathcal{G}' = \{G_j \in \mathcal{G} : Z \cap G_j = \emptyset\}$$

and

$$X' = \bigcup_{G_j \in \mathcal{G}'} G_j.$$

Then $(X', \mathcal{G}', \{B \backslash Z : Z \subseteq B \in \mathcal{B}\})$ is a $\mathrm{TD}_\lambda(t - i, k - i, v)$.

Theorem 1.16. *Let $(X, \mathcal{G}, \mathcal{B})$ be a $\mathrm{TD}_\lambda(t, k, v)$. Suppose $Y \subseteq X$ such that $|Y| = s \leq t$ and Y is a partial transversal of $\mathcal{G}$. Then there are exactly $\lambda_s = \lambda v^{t-s}$ blocks containing all the points in Y.*

Proof. Fix a subset of $t-s$ groups disjoint from Y, say $G'_1, \ldots, G'_{t-s}$. Consider a t-subset X consisting of all the points from Y and one point from each of $G'_1, \ldots, G'_{t-s}$. In particular, there are v^{t-s} such t-subsets X, and each such X occurs in precisely λ blocks. Note that every block that contains Y is a transversal of $\mathcal{G}$, so every such block contains exactly one such t-subset X. Therefore Y occurs in precisely λv^{t-s} blocks, as desired. □

Theorem 1.17. *Let $(X, \mathcal{G}, \mathcal{B})$ be a $\mathrm{TD}_\lambda(t, k, v)$. Suppose Y, $Z \subseteq X$ are group disjoint partial transversals of $\mathcal{G}$ such that $|Y| = i$, $|Z| = j$, and $i + j \leq t$. Then there are exactly*

$$\lambda_i^j = \lambda v^{t-i-j}(v-1)^j$$

blocks in $\mathcal{B}$ that contain all the points in Y and none of the points in Z.

Proof. Consider the set of groups G_Z that intersect Z. There are $(v-1)^j$ subsets X such that X consists of all the points from Y and one point from each group in G_Z, but X contains no points from Z. Each such $(i+j)$-subset X occurs in precisely λ_{i+j} blocks by Theorem 1.16. Therefore there are $\lambda_{i+j}(v-1)^j = \lambda v^{t-i-j}(v-1)^j$ blocks that contain all the points of Y but none of the points of Z. □

We can also apply the notion of *large sets* to transversal designs:

Definition 1.37. A *large set of* $\mathrm{TD}_\lambda(t, k, v)$ *on the point set X and group partition $\mathcal{G}$* is a set $\{(X, \mathcal{G}, \mathcal{B}_1), \ldots, (X, \mathcal{G}, \mathcal{B}_N)\}$ of $\mathrm{TD}_\lambda(t, k, v)$ in which every set of k points from distinct groups of X occurs as a block in precisely one of the $\mathcal{B}_i$s.

REMARK 1.12. It is easy to see that there must be $N = \frac{v^k}{\lambda v^t}$ transversal designs in a large set of $\mathrm{TD}_\lambda(t, k, v)$.

Transversal designs are equivalent to *orthogonal arrays*:

Definition 1.38. Let t, v, k, and λ be positive integers satisfying $k \geq t \geq 2$. An *orthogonal array* $\mathrm{OA}_\lambda(t, k, v)$ is a pair (X, D) such that the following properties are satisfied:

1. X is a set of v elements called *points*,
2. D is a λv^t by k array whose entries are elements of X, and
3. within any t columns of D, every t-tuple of points occurs in precisely λ rows.

EXAMPLE 1.12. An $\mathrm{OA}_1(2, 4, 3)$.

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 3 \\ 1 & 3 & 2 & 2 \\ 2 & 1 & 2 & 3 \\ 2 & 2 & 1 & 2 \\ 2 & 3 & 3 & 1 \\ 3 & 1 & 3 & 2 \\ 3 & 2 & 2 & 1 \\ 3 & 3 & 1 & 3 \end{array}$$

It is easy to see the correspondence between orthogonal arrays and transversal designs. Suppose (X, D) is an $\mathrm{OA}_\lambda(t, k, v)$. We define a bijection ϕ between the rows r_j of D and the blocks B_j of a $\mathrm{TD}_\lambda(t, k, v)$ as follows. For each row $r_j = [x_{j1}x_{j2}\cdots x_{jk}]$ of D, let

$$\phi(r_j) = \{(x_{j1}, 1), (x_{j2}, 2), \ldots (x_{jk}, k)\} = B_j$$

define a block B_j. Define $G_i = \{1, \ldots, v\} \times \{i\}$ for $1 \leq i \leq k$. Then $(X \times \{1, \ldots, k\}, \mathcal{G}, \mathcal{B})$ is a $\mathrm{TD}_\lambda(t, k, v)$ with $\mathcal{G} = \{G_i : 1 \leq i \leq k\}$ and $\mathcal{B} = \{B_j : 1 \leq j \leq \lambda v^t\}$.

EXAMPLE 1.13. The blocks of the $\mathrm{TD}_1(2, 4, 3)$ obtained from the $\mathrm{OA}_1(2, 4, 3)$ in Example 1.12:

$$\begin{array}{llllll} B_1\colon & (1,1) & (1,2) & (1,3) & (1,4) \\ B_2\colon & (1,1) & (2,2) & (3,3) & (3,4) \\ B_3\colon & (1,1) & (3,2) & (2,3) & (2,4) \\ B_4\colon & (2,1) & (1,2) & (2,3) & (3,4) \\ B_5\colon & (2,1) & (2,2) & (1,3) & (2,4) \\ B_6\colon & (2,1) & (3,2) & (3,3) & (1,4) \\ B_7\colon & (3,1) & (1,2) & (3,3) & (2,4) \\ B_8\colon & (3,1) & (2,2) & (2,3) & (1,4) \\ B_9\colon & (3,1) & (3,2) & (1,3) & (3,4) \end{array}$$

The above construction method can be reversed for an arbitrary $\mathrm{TD}_\lambda(t, k, v)$, say $(X, \mathcal{G}, \mathcal{B})$. To see this, note that we can relabel the points such that $X = \{1, \ldots, v\} \times \{1, \ldots, k\}$ and $\mathcal{G} = \{G_i : 1 \leq i \leq k\}$. Then the fact that any block and any group must contain exactly one common point implies that for each $B \in \mathcal{B}$, we can form the k-tuple $(b_1, \ldots, b_k)$, where $b_i \in B \cap G_i$ for $1 \leq i \leq k$. We can form an orthogonal array $\mathrm{OA}_\lambda(t, k, v)$ by taking all of these k-tuples as rows.

Definition 1.39. A *large set of* $\mathrm{OA}_\lambda(t,k,v)$ *on the point set* X is a set of $\mathrm{OA}_\lambda(t,k,v)$, say $\{(X,D_1),\ldots,(X,D_N)\}$, in which every k-tuple of elements from X occurs as a row in precisely one of the D_is. That is, the D_is form a partition of the set X^k of k-tuples with entries from X.

REMARK 1.13. It is easy to see that there must be $N = \frac{v^k}{\lambda v^t}$ orthogonal arrays in a large set of $\mathrm{OA}_\lambda(t,k,v)$.

A useful type of orthogonal array is a *linear array*, especially for constructing large sets:

Definition 1.40. Let (X,D) be an $\mathrm{OA}_\lambda(t,k,v)$. We say (X,D) is *linear* if $X = \mathbb{F}_q$ for some prime power q and the rows of D form a subspace of $(\mathbb{F}_q)^k$ of dimension $\log_q|D|$.

Linear orthogonal arrays (and hence the corresponding transversal designs) are easy to construct. In particular, the following is a useful construction method.

Theorem 1.18. *Suppose q is a prime power and k and ℓ are positive integers. Suppose M is an ℓ by k matrix over $\mathbb{F}_q$ such that every set of t columns of M is linearly independent. Then (X,D) is a linear $\mathrm{OA}_{q^{\ell-t}}(t,k,q)$, where D is the q^ℓ by k matrix formed by taking all linear combinations of the rows of M.*

Let q be a prime power and for every $x \in \mathbb{F}_q$, let $\vec{x} = [1, x, x^2, \ldots, x^{t-1}] \in (\mathbb{F}_q)^t$ for some integer $t \geq 2$. Construct the t by q matrix M by taking the columns to be the vectors $(\vec{x})^T$ for every $x \in \mathbb{F}_q$, where here $(\vec{x})^T$ means the transpose of $\vec{x}$. Applying Theorem 1.18 to M yields the following result:

Corollary 1.19. *Let $t \geq 2$ be an integer and let q be a prime power. Then there exists a linear $\mathrm{OA}_1(t,q,q)$.*

The following result is immediate.

Corollary 1.20. *Let $t \geq 2$ be an integer and let q be a prime power. Then there exists a linear $\mathrm{TD}_1(t,q,q)$.*

We now discuss how to construct a large set of linear orthogonal arrays from a "starting" linear orthogonal array. Suppose (X,D) is a linear $\mathrm{OA}_\lambda(t,k,v)$. We can obtain a large set of orthogonal arrays (and therefore transversal designs) from (X,D) by taking the set of cosets of D in $(\mathbb{F}_q)^k$. In particular, D is a subspace of $(\mathbb{F}_q)^k$, so the cosets of D form a partition of $(\mathbb{F}_q)^k$.

1.3. Thesis Outline

The remaining three chapters are devoted to our three research problems: Chapter 2 is on unconditionally secure signatures, Chapter 3 is on user-private information retrieval, and Chapter 4 is on the generalized Russian cards problem. A few elementary results from linear algebra, which are needed for the security proofs of our signature schemes, are included in Appendix A.

CHAPTER 2

Unconditionally Secure Signature Schemes Revisited

2.1. Introduction

Unconditionally secure signature (USS) schemes provide the ability to electronically sign documents without the reliance on computational assumptions needed in traditional digital signatures. That is, USS schemes are the analogue of digital signatures in the unconditionally secure cryptographic setting. The construction of such schemes is interesting not only from a theoretical perspective, but also from the viewpoint of ensuring security of information in the long term or designing schemes that are viable in a post-quantum world.

In traditional digital signatures, as mentioned in Section 1.1.1, each user has a pair consisting of a secret signing algorithm and a public verification algorithm. Since user verification algorithms are public, anyone can verify whether a given signature was created by the claimed signer. Unlike digital signatures, USS schemes require that verification algorithms are not public—for any possible signer, each user must have a different secret verification algorithm corresponding to that signer. The consequence is that USS schemes necessarily have a limited number of users, and hence a limited number of entities with the ability to verify a given signature, each with their own special test. Thus, any viable security definition for a USS scheme must carefully treat the subject of what constitutes a valid signature. That is, it is important to distinguish between signatures that are created using a user's signing algorithm and signatures that may satisfy one or more user verification algorithms. Current research [**38**–**40**, **58**, **65**] has proposed various models for unconditionally secure signature schemes, but these models do not fully treat the implications of having multiple verification algorithms or analyze the need for (and trust questions associated with) having a dispute resolution mechanism. We address both of these issues in this chapter.

Much of the material in this chapter appears in the paper "Unconditionally Secure Signature Schemes Revisited" [**76**], published in ICITS 2011.

Historically, there have been several attempts to create unconditionally secure constructions that satisfy security properties required for digital signatures, including non-repudiation, transferability, and unforgeability. Chaum and Roijakkers [8] introduced unconditionally secure signatures, proposing an interactive scheme that does not have transferability. Another approach to creating unconditionally secure signatures has been to enhance existing unconditionally secure message authentication codes (MACs), making these codes more robust in a signature setting. MACs clearly do not provide non-repudiation, as the sender and receiver compute authentication tags using the same algorithm. In addition, the need for a designated sender and receiver further limits the applicability of such schemes in a general signature setting.

Much research has been devoted to the removal of the standard MAC trust assumptions, in which both sender and receiver are assumed to be honest. In A^2-codes [43, 66, 67], the sender and receiver may be dishonest, but there is a trusted arbiter to resolve disputes; in A^3-codes [6,18,44], the arbiter is no longer trusted prior to dispute resolution, but is trusted to make an honest decision in the event of a disagreement. Johansson [44] used A^3-codes to improve the construction of Chaum and Roijakkers by making it non-interactive, but the signatures produced by the scheme are not transferable, as the use of a designated receiver limits the verification of the signature to those who have the appropriate key. Multi-receiver authentication codes (MRAs) [17] and multi-receiver authentication codes with dynamic sender (DMRAs) [59] use a broadcast setting to relax the requirement for designation of receivers, and also, in the latter case, senders. These codes are not appropriate outside of a broadcast setting, however, as neither non-repudiation nor transferability are satisfied.

Unsurprisingly, the first security models for unconditionally secure signature schemes, including Johansson [44] and Hanaoka et al. [38,39], drew upon the standard MAC security models. Shikata et al. [65] introduce a model using notions from public-key cryptography, which was later adopted in the work by Hara et al. [40] on blind signatures. Safavi-Naini et al. [58] present a MAC-based model meant to encompass the notions developed by Shikata et al. In this work, we present a new security model. Our model is more general than the MAC-based models of Hanaoka et al. [38, 39] and Safavi-Naini et al. [58] and covers the attacks described in these works. Like that of Shikata et al. [65], our work is based on security notions from traditional public-key signature systems. However, our model differs from those in the existing literature in its careful treatment of the concept of a "valid" signature. Our aim is to provide a rigorous and natural security model that covers all reasonable attacks.

In addition, we analyze a construction of Hanaoka et al. [38] in our model and provide a proof of security. We remark that while Hanaoka et al. make claims about the security of this construction in their model, they do not provide an analysis. In fact, security proofs are not provided for most of the constructions given in existing research. Thus, we feel it is useful to include our analysis of a basic unconditionally secure signature construction in our security model.

Our basic notion of security is easily extendable to a system with dispute resolution, which we argue is a necessary component of any USS scheme. Furthermore, our treatment of dispute resolution allows us to give formal definitions of non-repudiation and transferability. We show that a USS scheme that satisfies our unforgeability definition and has an appropriate dispute resolution method also satisfies non-repudiation and transferability, both of which are required properties for any reasonable signature scheme. Finally, we define various dispute resolution methods and examine the amount of trust each requires.

An advantage of our security framework for USS schemes is its flexibility; standard security properties from the literature, such as *strong key insulation* [**23**, **24**], can be incorporated into our basic model in a natural way. In key-insulated signature schemes, constructions are designed to be robust against signing-key exposure; this is done by splitting a user's signing information between a physically secure device (which stores the user's master key) and an insecure device (which is responsible for actually signing messages using temporary signing keys). We explore the notion of unconditionally secure strong key-insulated signatures in Sections 2.8 and 2.9, drawing from the work of Seito et al. [**63**] and Seito and Shikata [**62**] on unconditionally secure key-insulated multi-receiver authentication codes and key agreement. In particular, we give a formal extension of our security model to the strong key-insulation setting and present a construction that is secure in a restricted version of our model.

2.2. Overview of Contributions

The main contributions of this chapter are as follows.

- We present a new, natural security model for USS schemes based on security notions from traditional public-key signature systems. As such, our model is more general than the MAC-based models of Hanaoka et al. [**38**, **39**] and Safavi-Naini et al. [**58**]. Although Shikata et al. [**65**] also use notions from traditional public-key cryptography, our model differs from previous models in that it provides a rigorous treatment of the concept of "valid" signatures.
- We introduce the concept of *authentic*, *acceptable*, and *fraudulent* signatures. This new terminology allows us to distinguish between signatures produced by the signer's signing algorithm, signatures accepted by a verifier's verification algorithm, and signatures that are accepted by a verifier's verification algorithm but were not produced by the signer's signing algorithm.
- We define various dispute resolution methods and examine the amount of trust each requires. We show that under the *split trust assumption* normally given in the literature, typical suggested dispute resolution methods prove not to be sound. That is, we demonstrate the existence of a special type of forgery called a *dispute-enabled forgery* in these cases.
- We incorporate dispute resolution into our basic security model. We examine desirable properties for a dispute resolution mechanism and introduce the notion of

completeness, which implies authentic signatures are accepted by the dispute resolution method, and *soundness*, which implies that signatures that are not authentic are rejected by the dispute resolution method.
- We provide formal definitions of transferability and non-repudiation in the context of dispute resolution, and examine the relationship between these two properties and unforgeability.
- We analyze a construction of Hanaoka et al. [38] in our model and provide a proof of security.
- We define unconditionally secure signature schemes with key insulation, or KI-USS schemes, and extend our security model to this setting. Moreover, we present a construction that satisfies a restriction of our definition of unforgeability for key-insulated signature schemes, which is an extension of the notion of *strong key insulation* to the information-theoretically secure setting.

2.2.1. Chapter outline. In Section 2.3, we give a basic definition of a USS scheme, before moving to an informal treatment of the desired security properties. We then define a formal security model in Section 2.4. We introduce the notion of dispute resolution and give examples of possible dispute resolution methods in Section 2.5; we then formally define dispute resolution in Section 2.6 and explore the impact of dispute resolution on our basic security notions of unforgeability, non-repudiation, and transferability. We analyze the construction of Hanaoka et al. [38] in Section 2.7. In Sections 2.8 and 2.9, we give a formal treatment of USS schemes with strong key insulation and then we present our construction. In Section 2.10, we compare our work with that of previous literature. Finally, we give some concluding remarks in Section 2.11.

2.3. Preliminaries

We require the following definitions.

Definition 2.1. An unconditionally secure signature scheme (or USS scheme) Π consists of a tuple $(\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfying the following:

- The set $\mathcal{U} = \{U_1, \ldots U_n\}$ consists of n possible users, $\mathcal{X}$ is a finite set of possible messages, and Σ is a finite set of possible signatures.
- The *key-generation algorithm* Gen takes as input 1^k, where k is a security parameter, and outputs the signing algorithm Sign and the verification algorithm Vrfy.
- The *signing algorithm* $\mathsf{Sign}\colon \mathcal{X} \times \mathcal{U} \to \Sigma$ takes a message $x \in \mathcal{X}$ and a signer $U_\zeta \in \mathcal{U}$ as input, and outputs a signature $\sigma \in \Sigma$. For each $U_\zeta \in \mathcal{U}$, we let Sign_ζ denote the algorithm $\mathsf{Sign}(\cdot, U_\zeta)$.
- The *verification algorithm* $\mathsf{Vrfy}\colon \mathcal{X} \times \Sigma \times \mathcal{U} \times \mathcal{U} \to \{True, False\}$ takes as input a message $x \in \mathcal{X}$, a signature $\sigma \in \Sigma$, a signer $U_\zeta \in \mathcal{U}$, and a verifier $U_\nu \in \mathcal{U}$, and outputs either *True* or *False*. For each user U_ν, we let Vrfy_ν denote the algorithm $\mathsf{Vrfy}(\cdot, \cdot, \cdot, U_\nu)$.

It is required that, for every k, for every pair $(\mathsf{Sign}, \mathsf{Vrfy})$ output by $\mathsf{Gen}(1^k)$, for every pair U_ζ, $U_\nu \in \mathcal{U}$, and for every $x \in \mathcal{X}$, it holds that

$$\mathsf{Vrfy}_\nu\left(x, \mathsf{Sign}_\zeta(x), U_\zeta\right) = \mathit{True}.$$

REMARK 2.1. We are treating *deterministic* signature schemes only, in the sense that Sign and Vrfy are deterministic, although the above definition can easily be extended to the randomized setting. In practice, we typically also want Sign and Vrfy to be polynomial-time algorithms for efficiency. The point of USS schemes is to guarantee security against powerful adversaries, even those who are computationally unlimited.

We now define the concepts of authentic, acceptable, and fraudulent signatures. Distinguishing these three concepts is one of the main themes of this chapter.

Definition 2.2. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is ζ-*authentic* if $\sigma = \mathsf{Sign}_\zeta(x)$.

Definition 2.3. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is (ζ, ν)-*acceptable* if we have $\mathsf{Vrfy}_\nu(x, \sigma, U_\zeta) = \mathit{True}$.

Definition 2.4. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ is (ζ, ν)-*fraudulent* if σ is (ζ, ν)-acceptable but not ζ-authentic.

REMARK 2.2. In practice, we assume the existence of a trusted initializer TI, who takes responsibility for scheme set up and key distribution. That is, TI runs $\mathsf{Gen}(1^k)$ and securely distributes signing and verification keys to the appropriate users. Participants cannot create their own signing information and distribute corresponding verification keys to the other users, as in this case each user U_ζ would be able to create a (ζ, ν)-fraudulent signature for all $U_\nu \in \mathcal{U}$. While it might be possible to avoid this problem by using a "group computation" approach to create and distribute the necessary scheme information, for simplicity we assume the existence of a TI.

2.3.1. Security notions. Informally, a secure signature scheme should satisfy the following three properties:

1. *Unforgeability*: Except with negligible probability (with respect to the given security parameter k), it should not be possible to create a "valid" signature without the corresponding signing algorithm.
2. *Non-repudiation*: Except with negligible probability (with respect to the given security parameter k), a signer should be unable to repudiate a legitimate signature that he has created.
3. *Transferability*: Except with negligible probability (with respect to the given security parameter k), if a verifier accepts a signature, he can be confident that any other verifier will also accept it.

One objective of this paper is to formalize these notions in the unconditionally secure setting; we provide precise definitions in Sections 2.4 and 2.5. In contrast to the usual

public-key setting, the requirements of non-repudiation and transferability are not guaranteed in a USS scheme that satisfies the above intuitive notion of unforgeability. For "ordinary" digital signatures, non-repudiation is a consequence of unforgeability: a signature is considered "valid" if it passes a verification test, and it should be infeasible for anyone to create such a signature without knowledge of the secret signing algorithm. Thus, assuming the signing algorithm is not known to some third party, the signer cannot create a signature and later repudiate it. Transferability of digital signatures is guaranteed since there is a single, public verification algorithm.

In USS schemes, the concept of a "valid" signature requires clarification. Given sufficient computation time, a verifier is always capable of finding a signature that passes his own, secret verification test, so we cannot define the validity of a signature based on whether it passes a given user's verification algorithm. Indeed, there must be signatures that pass a given user's verification algorithm but that could not have been created with the signer's signing algorithm; otherwise the scheme does not satisfy unforgeability. Similarly, each verifier's verification algorithm must be different, or a given verifier may be able to present a signature acceptable to any verifier who possesses the same verification algorithm. A "valid" signature, then, must be created using the signer's signing algorithm, and it should be infeasible for anyone to create a signature that *appears* valid to other, non-colluding users, or the scheme does not have the properties of unforgeability, non-repudiation, and transferability. In particular, we have the following observations.

Theorem 2.1. *A necessary condition for a USS scheme to satisfy unforgeability is the existence of (ζ, ν)-fraudulent signatures for $\zeta \neq \nu$.*

Proof. Given sufficient computation time, a verifier U_ν can use his verification algorithm to create a (ζ, ν)-acceptable signature for any $\zeta \neq \nu$. If there are no (ζ, ν)-fraudulent signatures, then all signatures produced in this fashion must be ζ-authentic, and therefore they are successful forgeries. □

Theorem 2.2. *A USS scheme that satisfies unforgeability must also satisfy* $\mathsf{Vrfy}_\nu(\cdot,\cdot,\cdot) \neq \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$ *for* $\nu \neq \ell$.

Proof. Suppose that $\mathsf{Vrfy}_\nu(\cdot,\cdot,\cdot) = \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$ where $\nu \neq \ell$. Given sufficient computation time, U_ν can create a (ζ, ν)-acceptable signed message, (x, σ). Because $\mathsf{Vrfy}_\nu(\cdot,\cdot,\cdot) = \mathsf{Vrfy}_\ell(\cdot,\cdot,\cdot)$, it follows immediately that (x, σ) is (ζ, ℓ)-acceptable. This implies that the user U_ℓ will accept (x, σ) as a valid signature, but (x, σ) was not created by U_ζ. □

2.4. Formal Security Model

We now develop a formal security model for USS schemes. Our security definition is comparable to the notion of signatures secure against existential forgery under adaptive chosen message attacks in the case of public-key signature schemes. However, our definition takes into account the distinctive characteristics of the unconditional security setting, in

particular the existence (and necessity) of fraudulent signatures and multiple verification algorithms.

We specify two types of existential forgery. In our setting, an "existential" forgery is either a (ζ, ν)-fraudulent signature created without the help of the verifier U_ν, or a ζ-authentic signature created without the help of the signer U_ζ. If a USS scheme is secure, then both of these types of forgeries should be infeasible for an adversary to create.

We need the following oracles for our security definition:

- The $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ *oracle*; this oracle takes as input a message x and outputs an ℓ-authentic signature for the message x.
- The $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,\cdot)$ *oracle*; this oracle takes as input a signature pair (x, σ) and a signer U_ζ, and runs user U_ℓ's verification algorithm on input (x, σ, U_ζ), outputting *True* or *False*.

Definition 2.5. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k, let the set $C \subseteq \mathcal{U}$ be a coalition of at most ω users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* $\text{Sig-forge}_{C,\Pi}(k)$ with target signer U_ζ and verifier U_ν:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.
2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,U_\zeta)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively, with at most $\psi_S/(n - |C|)$ queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for each ℓ satisfying $U_\ell \notin C$. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$. We let $\mathcal{Q}$ denote the set of messages that the coalition submitted as queries to the oracles $\mathsf{Sign}_\zeta^{\mathcal{O}}(\cdot)$. Note that $\mathcal{Q}$ does not contain messages submitted as queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for $\ell \neq \zeta$.
3. The coalition C outputs a signature pair (x, σ).
4. The output of the game is defined to be 1 if and only if one of the following conditions is met:
   (a) $U_\nu \notin C$ and σ is a (ζ, ν)-fraudulent signature on x; or
   (b) $U_\zeta \notin C$, $x \notin \mathcal{Q}$, and σ is a ζ-authentic signature on x.

Definition 2.6. Let $\Pi = (\mathcal{U}, X, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\epsilon(k)$ be a negligible function of k. We say Π is $(\omega, \psi_S, \psi_V, \epsilon)$-*unforgeable* if for all coalitions C of at most ω possibly colluding users, and all choices of target signer U_ζ and verifier U_ν, it holds that

$$\Pr\left[\text{Sig-forge}_{C,\Pi}(k) = 1\right] \leq \epsilon(k).$$

REMARK 2.3. Another option is to include a $\mathsf{Fraud}_{(\zeta,\nu)}^{\mathcal{O}}(\cdot)$ *oracle*; this oracle takes as input a message x and outputs a (ζ, ν)-fraudulent signature on x. Providing certain (ζ, ν)-fraudulent signatures to the adversary could only increase his chances of ultimately constructing a new (ζ, ν)-fraudulent signature. Thus this would constitute a stronger security

model than the one we consider. On the other hand, it is hard to envisage a practical scenario where an adversary would have this kind of additional information about a verifier whom the adversary is attempting to deceive. Therefore we do not include the $\mathsf{Fraud}^{\mathcal{O}}$ oracle in our basic model of USS schemes. However, it would be straightforward to modify our model to include these oracles, if desired.

We observe that a scheme meeting the unforgeability requirement of Definition 2.6 satisfies our intuitive notions of non-repudiation and transferability. We explain these relationships in the following observations, noting that formal definitions of non-repudiation and transferability are intrinsically linked to the dispute resolution process, and so are provided later, in Section 2.5. We will formalize these observations in Theorems 2.9 and 2.10.

Observation 2.3. *An $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme Π provides non-repudiation.*

Proof. Suppose that Π is $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable. Then U_ζ cannot repudiate a given ζ-authentic signature σ, as Definition 2.6 guarantees that σ can be created without U_ζ only with negligible probability (as Condition 4(b) of Definition 2.5 holds only with negligible probability). Thus U_ζ cannot claim that other users may have created σ. The other possibility for a signer U_ζ to repudiate a signature on a message given to U_ν is if the signature is (ζ, ν)-fraudulent. Definition 2.6 also implies that U_ζ cannot create a (ζ, ν)-fraudulent signature (even with the help of $\omega - 1$ other users not including U_ν) except with negligible probability, as Condition 4(a) of Definition 2.5 is assumed to not hold (except with negligible probability). □

Observation 2.4. *An $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme Π provides transferability.*

Proof. In order for a signature σ to be non-transferable from U_ν to U_ℓ, the signature σ must be (ζ, ν)-acceptable, but not (ζ, ℓ)-acceptable, where $\nu \neq \ell$. If σ were ζ-authentic, it would also be (ζ, ℓ)-acceptable. Therefore σ must be (ζ, ν)-fraudulent. However, Definition 2.6 implies a (ζ, ν)-fraudulent signature cannot be created without the assistance of U_ν, except with negligible probability. □

From the point of view of a verifier, a scheme meeting Definition 2.6 gives reasonable assurance of the validity of a received signature. If a verifier U_ν receives a signature pair (x, σ) purportedly from U_ζ, then U_ν accepts the signature so long as σ is (ζ, ν)-acceptable for the message x. In this case, there are only two possibilities: either σ is ζ-authentic or (ζ, ν)-fraudulent for the message x. If σ is ζ-authentic, then a coalition that does not include the signer U_ζ has only a negligible probability of creating σ by Condition 4(b) of Definition 2.5. If σ is (ζ, ν)-fraudulent, then Condition 4(a) of Definition 2.5 guarantees that a coalition that does not include U_ν cannot create σ, except with negligible probability.

2.5. Dispute Resolution

Given that each verifier has his own distinct verification algorithm, a USS scheme must necessarily handle the event of a disagreement. That is, since there is no public verification method as in traditional digital signatures, a USS scheme must have a mechanism to determine the authenticity of a signature when some subset of users disagree whether a given signature should be accepted. In particular, dispute resolution is necessary to convince an outsider of the authenticity of a disputed signature. In traditional digital signatures, there are no outsiders to the scheme, in the sense that everyone has access to the public verification method. In our setting, however, the number of participants (and therefore their access to verification algorithms) is limited. Dispute resolution is a method that effectively deals with the need for resolution of disagreements in, for example, a court setting. Typically, dispute resolution involves all the users voting on the validity of a signature, or alternatively, a trusted arbiter stating whether a signature is valid.

The manner in which a dispute resolution mechanism may be invoked necessarily affects the security of the overall scheme. In particular, we should not allow users to invoke the dispute resolution mechanism an arbitrary number of times. If users have unlimited access, it may be possible for a coalition to use dispute resolution as a type of verification oracle against a target signer U_ζ. As this is undesirable, we need to limit access to dispute resolution in a reasonable way. One simple possibility is to limit dispute resolution to once per scheme. That is, once dispute resolution has been invoked, we require that the users request the TI to generate new signing and verification keys. This may be reasonable because dispute resolution necessarily implies the existence of a lying (or otherwise compromised) user, and we find it unlikely that users will want to continue the current scheme with the dishonest (or compromised) user in question; we discuss these concepts in more detail in Remark 2.4. That said, it may be desirable to include a mechanism by which to determine and punish cheaters—such a mechanism may be useful if multiple calls to the dispute resolution are desired, or to determine which users should not be included in a scheme reset.

We focus on the case in which dispute resolution causes a scheme reset. We begin with some basic concepts and then provide and analyze examples of possible dispute resolution mechanisms. Ideally, the dispute resolution process validates a signature if and only if the signature is authentic, i.e., the signature was produced by the purported signer. This leads to the following definitions.

Definition 2.7. A *dispute resolution method* $\mathcal{DR}$ for a USS scheme Π is a procedure invoked when a pair of users U_ℓ, $U_{\ell'} \in \mathcal{U}$ disagrees as to the validity of a given signature (x, σ), purportedly signed by U_ζ. Here U_ℓ (respectively, $U_{\ell'}$) may be any user in $\mathcal{U}$, including U_ζ. The procedure $\mathcal{DR}$ consists of an algorithm DR that takes as input a signature pair (x, σ) and a purported signer U_ζ, and outputs a value in $\{Valid, Invalid\}$, subject to the following rules:

1. If DR outputs *Valid*, then (x, σ) must subsequently be accepted as a ζ-authentic signature on x by all users.
2. If DR outputs *Invalid*, then (x, σ) must subsequently be rejected by all users.

We remark that the algorithm DR may have access to additional (secret) scheme information, as specified by the particular dispute resolution method.

The following definitions capture the desirable properties of a given $\mathcal{DR}$.

Definition 2.8. *Soundness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say $\mathcal{DR}$ is *sound* if, whenever σ is not a ζ-authentic signature on x, then $\mathsf{DR}\big((x, \sigma), U_\zeta\big)$ outputs *Invalid*.

Definition 2.9. *Completeness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say $\mathcal{DR}$ is *complete* if, whenever σ is a ζ-authentic signature on x, then $\mathsf{DR}\big((x, \sigma), U_\zeta\big)$ outputs *Valid*.

Definition 2.10. *Correctness.* Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. If $\mathcal{DR}$ is both sound and complete, we say $\mathcal{DR}$ is *correct*.

Remark 2.4. A correct dispute resolution method $\mathcal{DR}$ is useful in terms of identifying and punishing users who are cheating (or alternatively whose secret information has been compromised). To see this, suppose that a signature σ with purported signer U_ζ is given to $\mathcal{DR}$ by two users U_ℓ and $U_{\ell'}$. Without loss of generality, suppose U_ℓ claims σ should be accepted and $U_{\ell'}$ claims σ should be rejected. Then if the output of $\mathcal{DR}$ is *Valid*, soundness implies that σ is ζ-authentic. In this case, the user $U_{\ell'}$ is either dishonest or otherwise compromised. If, on the other hand, the output of $\mathcal{DR}$ is *Invalid*, completeness implies that σ is not ζ-authentic. In this case, the user U_ℓ is either dishonest or otherwise compromised. Here, by *otherwise compromised*, we are recognizing the possibility that a user's secret information may become *unintentionally* known to an adversary (i.e., a coalition of dishonest users), but the user in question is honest. This might happen, for example, due to insecure storage of signing and/or verification keys.

We define three dispute resolution methods and examine the level of honesty required in each scheme. In particular, we wish to define trust assumptions sufficient to ensure the correctness of these dispute resolution methods. That is, we consider the degree of trust a group of users should have in order to use a particular dispute resolution method.

Definition 2.11. We have the following dispute resolution methods, assuming a disputed signature σ on message x with purported signer U_ζ:

- *Omniscient Arbiter (OA) Dispute Resolution*: Designate an arbiter equipped with all of the USS scheme set-up information. The signature σ is considered valid if the arbiter, using his knowledge of all the signing and verification algorithms, accepts the signature as authentic. Here we assume the arbiter is honest.

- *Verifier-equivalent Arbiter (VEA) Dispute Resolution*: Designate an arbiter equipped with his own verification algorithm, $\mathsf{Vrfy}_{\mathcal{A}}$, (i.e., the arbiter is a *glorified verifier*). The arbiter tests the authenticity of the signature σ by running $\mathsf{Vrfy}_{\mathcal{A}}(x, \sigma, U_\zeta)$; the signature is considered valid if $\mathsf{Vrfy}_{\mathcal{A}}(x, \sigma, U_\zeta)$ outputs *True*. Here we assume the arbiter is honest. We remark that the arbiter may or may not be a normal user in the scheme, although assuming the arbiter is honest may be more reasonable if the arbiter is not otherwise involved with the scheme.
- *Majority Vote (MV) Dispute Resolution*: Here we resolve disputes by having the users vote on the validity of the signature σ. Each user is responsible for running his verification algorithm on (x, σ, U_ζ) and casting a *valid vote* if his verification algorithm outputs *True* and an *invalid vote* otherwise. The signature is considered valid if a prespecified threshold of *valid* votes are cast; here we consider the case of a majority threshold and assume all users vote. We assume that a majority of users are honest.

In the case of OA dispute resolution, it is clear that we require the arbiter to be honest, as he has all the necessary information to sign and verify documents on behalf of other users. That is, a USS scheme Π with OA dispute resolution clearly cannot satisfy any unforgeability condition unless the arbiter is honest, as the arbiter has all the necessary information to sign messages on behalf of users. Moreover, provided that the arbiter is honest, this dispute resolution method is both sound and complete, as the arbiter is able to determine the authenticity of a given signature and behave appropriately. In fact, the correctness of OA dispute resolution with an honest arbiter is independent of the security of the underlying scheme. Correctness implies the arbiter's ability to identify signatures that are ζ-authentic for the purported signer U_ζ, which is independent from the problem of preventing other users from creating a ζ-authentic signature without U_ζ's help. However, it is of course still the case that correct dispute resolution is only useful in conjunction with an unforgeable USS scheme. To summarize, we have the following result:

Theorem 2.5. *Let Π be a USS scheme and let $\mathcal{DR}$ be an OA dispute resolution method for Π with an honest arbiter. Then $\mathcal{DR}$ is correct.*

Remark 2.5. Although in this chapter we focus on deterministic signature schemes, an interesting observation with respect to OA dispute resolution arises in the case of randomized signature schemes. In particular, if the signature scheme is randomized, then the arbiter may have to be computationally unbounded in order to perform dispute resolution. That is, if a purported signer claims a disputed signature is not valid, the arbiter may have to search an exponential space. This issue does not arise if the purported signer does not dispute the validity of the signature, however, as in this case the signer can simply reveal the randomness used to produce the disputed signature.

In the following two theorems, we present trust assumptions sufficient to achieve correctness in VEA and MV dispute resolution. For these methods, it is necessary to consider the security properties of the underlying signature scheme Π.

Theorem 2.6. *Let* Π *be an* $(\omega, \psi_S, \psi_V, \epsilon)$*-unforgeable USS scheme and let* $\mathcal{DR}$ *be a VEA dispute resolution method for* Π *with an honest arbiter. Then* $\mathcal{DR}$ *is correct in the presence of a coalition of users of maximum size* ω*, except with negligible probability.*

Proof. Suppose we have a disputed signature σ on message x with purported signer U_ζ. The arbiter $\mathcal{A}$ outputs *Valid* if and only if σ is $(\zeta, \mathcal{A})$-acceptable.

Given that the underlying scheme Π satisfies our unforgeability definition, a coalition of maximum size ω cannot produce a signature that is $(\zeta, \mathcal{A})$-fraudulent without $\mathcal{A}$'s help, except with negligible probability. That is, an honest arbiter $\mathcal{A}$ outputs *Valid* exactly when σ is ζ-authentic (except with negligible probability). □

Theorem 2.7. *Let* Π *be an* $(\omega, \psi_S, \psi_V, \epsilon)$*-unforgeable USS scheme and let* $\mathcal{DR}$ *be an MV dispute resolution method for* Π*. Then* $\mathcal{DR}$ *is correct in the presence of a coalition of dishonest users of maximum size* $\min\left\{\omega, \left\lfloor \frac{n-1}{2} \right\rfloor\right\}$*, except with negligible probability.*

Proof. Suppose we have a disputed signature σ on message x with purported signer U_ζ. Consider a coalition C of size at most $\min\left\{\omega, \left\lfloor \frac{n-1}{2} \right\rfloor\right\}$. If x is ζ-authentic, then any honest $U_\ell \notin C$ will cast a *valid* vote. The coalition C can attempt to ensure that x is rejected by having each member cast an *invalid* vote, but as long as a majority of users are honest, x will be accepted by the dispute resolution process. If x is not ζ-authentic, then since Π is $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable, we have that x is not (ζ, ℓ)-fraudulent for any honest $U_\ell \notin C$ except with negligible probability. That is, any honest U_ℓ will (with overwhelming probability) cast a *invalid* vote. The members of C can attempt to have x accepted by having each member cast a *valid* vote, but given that a majority of users are honest, this approach works with only negligible probability. □

Remark 2.6. The proof of Theorem 2.6 establishes that the correctness of the VEA dispute resolution method depends on how easy it is in the underlying scheme Π to construct signatures which are $(\zeta, \mathcal{A})$-acceptable for users $U_\zeta \in \mathcal{U}$. In particular, it is easy to see that if Π does not satisfy Definition 2.6 for some ω with respect to output 4(a) of the security game Sig-forge$_{C,\Pi}(k)$ (as defined in Definition 2.5), then the VEA method fails to be correct, even with an honest arbiter. In addition, if we consider the maximum ω for which Π satisfies the above unforgeability criterion, it is easy to see that the VEA method fails to be correct in the presence of more than ω colluding users. Similar observations hold for Theorem 2.7 with respect to the MV dispute resolution method.

As observed above, we achieve correctness of the VEA dispute resolution method by assuming that the arbiter is honest. Achieving soundness and completeness is not as clear if we weaken this honesty requirement, however. In the typical VEA dispute resolution methods considered in current literature [**40**, **58**, **65**], the arbiter is assumed to be a glorified verifier, with the same type of keying information as an arbitrary verifier. The arbiter is assumed to follow the rules of the dispute resolution method honestly and is otherwise treated as a normal user in the context of the security model, i.e., he is allowed to be

dishonest otherwise. That is, the arbiter is allowed to be a member of the coalition attempting to create a forgery, but he is expected to follow the dispute resolution process itself honestly. We refer to this set of trust assumptions as the *split trust assumption*. We argue that the split trust assumption is problematic, however, and should likely be abandoned. In particular, if we consider VEA dispute resolution where we allow the arbiter to be part of a given coalition, then soundness is no longer guaranteed.

The arbiter's distinct role in the dispute resolution method necessitates a more careful study of the arbiter, and therefore treating the arbiter as a normal verifier in the context of the security model is insufficient. While it is obvious an arbiter who is dishonest during dispute resolution can cause a fraudulent signature to be deemed valid, we cannot allow the arbiter to be dishonest before dispute resolution either, contrary to the claims of Safavi-Naini et al. [58] and Shikata et al. [65]. In particular, the VEA dispute resolution method does not achieve soundness under the split trust assumption due to the existence of a new type of forgery introduced by the dispute resolution process, which we term a *dispute-enabled forgery*:

Definition 2.12. Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We say a signature σ on a message $x \in \mathcal{X}$ is a *dispute-enabled forgery for signer* U_ζ if σ is not ζ-authentic, but $\mathsf{DR}\big((x,\sigma), U_\zeta\big)$ outputs *Valid*.

In fact, the proof of Theorem 2.6 indicates why the split trust assumption is problematic: an honest arbiter $\mathcal{A}$ outputs *Valid* during dispute resolution if and only if the signature is $(\zeta, \mathcal{A})$-acceptable for purported signer U_ζ. But if we allow $\mathcal{A}$ to be dishonest prior to dispute resolution, then $\mathcal{A}$ can produce a signature x that is $(\zeta, \mathcal{A})$-fraudulent. In this case, $\mathcal{A}$'s verification algorithm outputs *True* on input x with signer U_ζ, so x is a dispute-enabled forgery. We remark that the case of MV may be viewed as a generalized version of VEA dispute resolution and the security concerns are similar.

The main observation is that a cheating arbiter $\mathcal{A}$ (or, in the case of MV dispute resolution, a collusion of a majority of verifiers) can successfully forge a (ζ, ν)-fraudulent signature for any cooperating user U_ν. Hence, VEA and MV dispute resolution do not protect the signer against a dishonest arbiter (or a dishonest majority of verifiers) *under the split trust assumption*, since dispute-enabled forgeries exist. From the perspective of signer security, the split trust assumption is certainly not reasonable.

From the perspective of verifier security, it is interesting to note that both the VEA and MV methods are acceptable under the split trust assumption. This is a consequence of the fact that both the VEA and MV methods are complete provided that the dispute resolution process itself is performed honestly. We show in Theorems 2.9 and 2.10 that completeness is sufficient for an $(\omega, \psi_S, \psi_V, \epsilon)$-USS scheme Π with dispute resolution $\mathcal{DR}$ to provide non-repudiation and transferability. That is, the VEA and MV methods do not require the arbiter(s) to be honest prior to dispute resolution in order to achieve non-repudiation and transferability. As seen above, however, the VEA and MV methods require the arbiter(s)

to be honest prior to dispute resolution in order to achieve soundness. In this sense, we see that VEA and MV dispute resolution under the split trust assumption provide similar *verifier security* to OA dispute resolution with an honest arbiter (in that non-repudiation and transferability are assured), but they fail to provide similar *signer security* (in that unforgeability is not assured).

Nonetheless, we argue that a more reasonable approach to dispute resolution is to assume the possibility of cheating both *before and during* dispute resolution. In this case, we see that for the VEA method, we must have an honest arbiter $\mathcal{A}$, and for the MV method, we require that a majority of users are honest.

With these examples in mind, we give a formal treatment of dispute resolution in the following section.

2.6. A Formal Treatment of Dispute Resolution

The possibility of dispute-enabled forgeries requires an extension to the unforgeability requirement of a USS scheme. Although unforgeability (unlike non-repudiation and transferability) is not intrinsically linked to the dispute resolution process, we need to ensure that the dispute resolution process itself does not weaken the overall security of the scheme.

Definition 2.13. Let Π be a USS scheme and let $\mathcal{DR}$ be a dispute resolution method for Π. We extend the signature game Sig-forge$_{C,\Pi}(k)$ to the *signature game* $\mathcal{DR}$-Sig-forge$_{C,\Pi}(k)$ by adjusting Definition 2.5 as follows. We make the following changes to Step 4:

4. We add the following to the list of possible conditions for which the output of the game is 1:
   (c) $U_\zeta \notin C$, σ is not ζ-authentic, but $\mathsf{DR}\big((x,\sigma),U_\zeta\big)$ outputs *Valid*.

Definition 2.14. Let $\Pi = (\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let $\epsilon(k)$ be a negligible function of k. We say the pair $(\Pi, \mathcal{DR})$ is $\mathcal{DR}$-*unforgeable* with parameters $(\omega, \psi_S, \psi_V, \epsilon)$ if for all coalitions C of at most ω possibly colluding users, and all choices of target signer U_ζ and verifier U_ν, it holds that

$$\Pr[\mathcal{DR}\text{-Sig-forge}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

Remark 2.7. Here we model an attack in which a call to dispute resolution necessitates an immediate scheme reset. If we wish to account for the possibility of multiple calls to the dispute resolution method, we can allow C bounded access to a new oracle, the $\mathcal{DR}(\cdot,\cdot,\cdot)$ *oracle*, which takes as input a signature pair (x,σ) and a signer U_ζ and simulates the dispute resolution method $\mathcal{DR}$ on input (x,σ,U_ζ), outputting either *Valid* or *Invalid*.

We now observe that given an underlying scheme Π satisfying our *original* definition of unforgeability (Definition 2.6), we can achieve our stronger definition of $\mathcal{DR}$-unforgeability by choosing a *sound* dispute resolution method $\mathcal{DR}$.

Theorem 2.8. *Let* Π *be an* $(\omega, \psi_S, \psi_V, \epsilon)$*-unforgeable USS scheme and let* $\mathcal{DR}$ *be a sound dispute resolution method for* Π*. Then the pair* $(\Pi, \mathcal{DR})$ *is* $\mathcal{DR}$-unforgeable *with parameters* $(\omega, \psi_S, \psi_V, \epsilon)$*.*

Proof. Since Π is $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable, we see that a coalition C of at most ω users cannot produce signatures satisfying Conditions 4(a) or 4(b) of Definition 2.13 except with negligible probability. If, in addition, the dispute resolution method $\mathcal{DR}$ is sound, then $\mathcal{DR}$ outputs *Invalid* when given a signature that is not ζ-authentic for (any choice of) target signer U_ζ, so C cannot produce a signature satisfying Condition 4(c). □

Remark 2.8. Condition 4(c) of Definition 2.13 says that a possible successful output of the game $\mathcal{DR}$-Sig-forge$_{C,\Pi}(k)$ is a dispute-enabled forgery. Definition 2.14 implies that with high probability, the coalition C is unable to find a dispute-enabled forgery. In other words, the coalition C is able to produce a signature compromising the soundness of the dispute resolution method $\mathcal{DR}$ with only negligible probability.

We now discuss the properties of non-repudiation and transferability. As previously mentioned, both of these properties are intrinsically linked to dispute resolution. That is, the outcome of the chosen dispute resolution method determines the success or failure of these attacks. In particular, we show that completeness is sufficient to achieve both non-repudiation and transferability.

We remark that in order for the dispute resolution method to be invoked in the first place, there must be disagreement as to the validity of a given signature σ. In a *repudiation attack*, the signer U_ζ gives a (ζ, ν)-acceptable signature σ to the verifier U_ν (i.e., σ appears valid to U_ν) and then later denies the validity of σ. In this case, the signer U_ζ and the target verifier U_ν will invoke the dispute resolution method. Similarly, for a *transferability attack*, a verifier U_ν transfers a signature σ that is (ζ, ν)-acceptable (i.e., σ appears valid to U_ν) to another user U_ℓ, who rejects σ as invalid. Thus, the dispute resolution method is again invoked, this time by users U_ν and U_ℓ. In this case, U_ν is assumed to be honest, but we remark that it is also possible that U_ℓ is honest, in the sense that U_ℓ may genuinely believe the signature in question to be invalid. That said, it is also possible for U_ℓ to be part of the attempt to "trap" U_ν (independently of whether or not the given signature is rejected by U_ℓ's verification algorithm). We now provide formal definitions of these two attacks.

Definition 2.15. Let $\Pi = (\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let the set $C \subseteq \mathcal{U}$ be a coalition of at most ω users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* Repudiation$_{C,\Pi}(k)$ with signer $U_\zeta \in C$ and target verifier U_ν satisfying $U_\nu \notin C$:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.
2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot, \cdot, U_\zeta)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the

$\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively, with at most $\psi_S/(n - |C|)$ queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for each ℓ satisfying $U_\ell \notin C$. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$.

3. The coalition C outputs a signature pair (x, σ).
4. The output of the game is defined to be 1 if and only if one of the following conditions are met:
   (a) σ is (ζ, ν)-fraudulent and the dispute resolution method $\mathcal{DR}$ (as invoked by U_ζ and U_ν) rejects σ as *Invalid*.
   (b) σ is ζ-authentic and the dispute resolution method $\mathcal{DR}$ (as invoked by U_ζ and U_ν) rejects σ as *Invalid*.

Definition 2.16. Let $\Pi = (\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let $\epsilon(k)$ be a negligible function of k. We say the combined scheme $(\Pi, \mathcal{DR})$ satisfies *non-repudiation* with parameters $(\omega, \psi_S, \psi_V, \epsilon)$ if for all coalitions C of at most ω possibly colluding users, and for all choices of signer U_ζ and target verifier U_ν, it holds that

$$\Pr[\text{Repudiation}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

In the following theorem, we demonstrate that a dispute resolution method $\mathcal{DR}$ that is complete, when combined with a underlying USS scheme Π that is unforgeable, suffices to ensure non-repudiation attacks are (highly) unlikely to succeed.

Theorem 2.9. *Let Π be an $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme and let $\mathcal{DR}$ be a complete dispute resolution method for Π. Then $(\Pi, \mathcal{DR})$ provides non-repudiation.*

Proof. Assume Π does not provide non-repudiation; that is, the game $\text{Repudiation}_{C,\Pi}(k)$ outputs 1 with non-negligible probability. Suppose $\text{Repudiation}_{C,\Pi}(k)$ with signer U_ζ and target verifier U_ν outputs 1. Then C has created a (ζ, ν)-acceptable signature pair (x, σ), such that the dispute resolution method $\mathcal{DR}$ (as invoked by U_ζ and U_ν) rejects σ as *Invalid*.

Now, σ is either ζ-authentic or (ζ, ν)-fraudulent. If σ is (ζ, ν)-fraudulent, then Condition 4(a) of Definition 2.5 holds, so the output of $\text{Sig-forge}_{C,\Pi}(k)$ with target signer $U_\zeta \in C$ and verifier $U_\nu \notin C$ is 1 (with non-negligible probability). That is, Π is not $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable. If σ is ζ-authentic, then $\mathcal{DR}$ rejected a ζ-authentic signature and therefore the dispute resolution method is not complete. □

Definition 2.17. Let $\Pi = (\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let the set $C \subseteq \mathcal{U}$ be a coalition of at most ω users, and let ψ_S and ψ_V be positive integers. We define the following *signature game* $\text{Non-transfer}_{C,\Pi}(k)$ with signer U_ζ and target verifier U_ν, where $U_\nu \notin C$:

1. $\mathsf{Gen}(1^k)$ is run to obtain the pair $(\mathsf{Sign}, \mathsf{Vrfy})$.

2. The coalition C is given bounded access to the oracles $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,U_\zeta)$ for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively, with at most $\psi_S/(n-|C|)$ queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for each ℓ satisfying $U_\ell \notin C$. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$.
3. The coalition C outputs a signature pair (x, σ).
4. The output of the game is defined to be 1 if and only if the following conditions are met:
   (a) σ is (ζ,ν)-fraudulent and the dispute resolution method $\mathcal{DR}$, as invoked by U_ν and some user $U_\ell \in \mathcal{U}$, outputs *Invalid*.
   (b) σ is ζ-authentic and the dispute resolution method $\mathcal{DR}$, as invoked by U_ν and some verifier $U_\ell \in C$, outputs *Invalid*.

REMARK 2.9. The distinction between the two cases in part 4 of Definition 2.17 is with respect to the integrity of the users who invoke the dispute resolution method. In the first case, it is possible that an honest verifier $U_\ell \notin C$ for whom σ is not (ζ,ℓ)-fraudulent may be involved, hence (unwittingly) aiding the coalition in trapping the target verifier U_ν. If σ is ζ-authentic, then there is no such user, as all honest verifiers would accept σ, so a member of the coalition C must participate in invoking dispute resolution.

Definition 2.18. Let $\Pi = (\mathcal{U}, \mathcal{X}, \Sigma, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a USS scheme with security parameter k and let $\mathcal{DR}$ be a dispute resolution method for Π. Let $\epsilon(k)$ be a negligible function of k. We say the combined scheme $(\Pi, \mathcal{DR})$ satisfies *transferability* with parameters $(\omega, \psi_S, \psi_V, \epsilon)$ if for all choices of signer U_ζ and target verifier U_ν, it holds that

$$\Pr[\text{Non-transfer}_{C,\Pi}(k) = 1] \leq \epsilon(k).$$

The following theorem is similar to Theorem 2.9 and gives the corresponding result for transferability.

Theorem 2.10. *Let Π be an $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable USS scheme and let $\mathcal{DR}$ be a complete dispute resolution method for Π. Then $(\Pi, \mathcal{DR})$ satisfies transferability.*

Proof. Suppose Π does not provide transferability and assume the game $\text{Non-transfer}_{C,\Pi}(k)$ outputs 1, with signer U_ζ and target verifier $U_\nu \notin C$. Then C output a signature pair (x, σ) such that σ is (ζ,ν)-acceptable and the dispute resolution method (as invoked by U_ν and some user U_ℓ) rejected σ as *Invalid*.

Now, σ is either (ζ,ν)-fraudulent or ζ-authentic. If the former holds, then Condition 4(a) of Definition 2.5 is met. That is, the output of $\text{Sig-forge}_{C,\Pi}(k)$ with target signer U_ζ and verifier U_ν is 1 (with non-negligible probability), so Π is not $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable. If the latter holds, then the dispute resolution method rejected a ζ-authentic signature and is therefore not complete. □

Together, Theorems 2.8, 2.9, and 2.10 provide sufficient conditions for a USS scheme Π and a dispute resolution method $\mathcal{DR}$ to satisfy the desired properties of unforgeability, non-repudiation, and transferability. In particular, it suffices to take Π to be $(\omega, \psi_S, \psi_V, \epsilon)$-unforgeable and $\mathcal{DR}$ to be sound and complete (i.e., correct). Furthermore, we remark that Condition 4b of Definition 2.15 and Condition 4b of Definition 2.17 both correspond to a demonstration of the lack of completeness of the associated $\mathcal{DR}$. That is, in a scheme that satisfies non-repudiation or transferability, it must be infeasible to find a signature pair that acts as a witness to the lack of completeness of the associated $\mathcal{DR}$.

2.7. Basic USS Scheme Construction and Analysis

Current literature favors constructions using multivariate polynomials. We consider the security of the construction from Hanaoka et al. [38] in our security model. We reiterate that Hanaoka et al. [38] do not provide a proof of security for this construction in their model.

2.7.1. Key pair generation. Let $\mathbb{F}_q$ be a finite field with q elements such that $q > n$. (In practice, we pick q to be much larger than n.) The TI picks n *verification vectors* $\vec{v_1}, \ldots, \vec{v_n} \in (\mathbb{F}_q)^\omega$ uniformly at random for users $U_1, \ldots, U_n$, respectively, subject to one additional constraint. For technical reasons, we assume the verification vectors $\vec{v_1}, \ldots, \vec{v_n} \in (\mathbb{F}_q)^\omega$ satisfy the additional property that for any subset of size $\omega + 1$, the corresponding subset of size $\omega + 1$ formed from the new vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. (This linear independence assumption is used in the security proof in Section 2.7.3.) We assume user identities $U_1, \ldots, U_n$ have a representation as elements in $\mathbb{F}_q$ in some suitable (and public) way.

The TI constructs the polynomial $F(x, y_1, \ldots, y_\omega, z)$ as

$$F(x, y_1, \ldots, y_\omega, z) = \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} a_{i0k} x^i z^k + \sum_{i=0}^{n-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} a_{ijk} x^i y_j z^k,$$

where the coefficients $a_{ijk} \in \mathbb{F}_q$ are chosen uniformly at random.

For each user U_ζ for $1 \leq \zeta \leq n$, the TI computes the *signing key* $s_\zeta(y_1, \ldots, y_\omega, z) = F(x, y_1, \ldots, y_\omega, z)|_{x=U_\zeta}$ and the *verification key* $\tilde{v}_\zeta(x, z) = F(x, y_1, \ldots, y_\omega, z)|_{(y_1,\ldots,y_\omega)=\vec{v_\zeta}}$. For each user, the TI distributes the verification vector $\vec{v_\zeta}$, the signing key $s_\zeta(y_1, \ldots, y_\omega, z)$, and the verification key $\tilde{v}_\zeta(x, z)$ to the corresponding user $U_\zeta \in \mathcal{U}$. It is assumed the TI can communicate with the users via secure channels and deletes the information afterwards.

2.7.2. Signature generation and verification. For a message $m \in \mathbb{F}_q$, a user U_ζ generates a signature σ by

$$\sigma(y_1, \ldots, y_\omega) = s_\zeta(y_1, \ldots, y_\omega, z)|_{z=m}.$$

To verify a signature pair (m, σ) from U_ζ, a user U_ν checks that

$$\sigma(y_1, \ldots, y_\omega)|_{(y_1,\ldots,y_\omega)=\vec{v_\nu}} = \tilde{v}_\nu(x, z)|_{x=U_\zeta, z=m}.$$

REMARK 2.10. The parameter ω in the construction determines the maximum number of colluders the scheme protects against and the parameter ψ determines the maximum number of signatures each user can produce without revealing their signing information. This is discussed in detail in the security analysis, but for clarity we briefly sketch how the construction relates to these bounds. In particular, each signing key s_ζ is a polynomial of degree ψ in z, so users cannot produce more than ψ signatures without revealing s_ζ. In addition, if a coalition C consists of $\omega+1$ users or more, then the verification keys $\{\tilde{v}_h\}_{U_h \in C}$ suffice to reconstruct F. This follows because F is a linear polynomial in $\mathbb{F}_q[x, z][y_1, \ldots, y_\omega]$, and each verification key is a point on $F(y_1, \ldots, y_\omega)$. In this case the coalition has $\omega + 1$ linearly independent linear equations in $\omega + 1$ unknowns, and so C can solve for F.

2.7.3. Security analysis. Given q, we define the security parameter to be k, where $k = \log_2 q$. We consider the game Sig-forge$_{C,\Pi}(k)$ and calculate the probability that the output is 1. In particular, we consider the probability that the coalition C produces a signature pair (m, σ) satisfying Conditions 4(a) and 4(b) of Definition 2.5 separately. Here we prove the scheme is unforgeable with respect to coalitions C of size at most ω, where C is allowed $\psi_S = (n - \omega)\psi$ oracle queries to $\mathsf{Sign}^{\mathcal{O}}$ (where ψ is the total number of $\mathsf{Sign}_h^{\mathcal{O}}$ oracle queries allowed for each user $U_h \notin C$), and where the number of $\mathsf{Vrfy}^{\mathcal{O}}$ queries, say ψ_V, is arbitrary. (As shown in the following theorem, the probability that C creates a successful forgery depends on this value ψ_V.) That is, we allow C to have at most ω members and to have access to ψ sample signatures from each user $U_h \notin C$. (This is consistent with the fact that in this USS scheme, each user is allow to produce at most ψ signatures, so the bound on oracle access to $\mathsf{Sign}_h^{\mathcal{O}}$ for each user $U_h \notin C$ must be ψ.)

Theorem 2.11. *Under the above assumptions, C outputs a signature pair (m, σ) in the game* Sig-forge$_{C,\Pi}(k)$ *of Definition 2.5 satisfying Condition 4(a) with probability at most $\frac{1}{q-\psi_V-1}$ and Condition 4(b) with probability at most $\frac{1}{q-\psi_V}$.*

Proof. Recall the assumption that the verification vectors $\vec{v_1}, \ldots, \vec{v_n} \in (\mathbb{F}_q)^\omega$ satisfy the additional property that for any subset of size $\omega+1$, the corresponding subset of size $\omega+1$ formed from the new vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. We use this fact throughout the proof.

We wish to consider the strongest possible coalition C. To this end, we consider a coalition of size ω whose verification vectors form a linearly independent set. Lemma A.1 implies that this is always possible. Without loss of generality, assume our adversaries are

$C = \{U_1, \ldots, U_\omega\}$, with target signer U_ζ and target verifier U_ν. The coalition C outputs a signature pair (m, σ) with claimed signer U_ζ.

For ease of notation, define $y_0 = 1$ and let $\vec{y}$ denote the vector $(y_0, y_1, \ldots, y_\omega)$. Then we have

$$F(x, \vec{y}, z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{ijk} x^i y_j z^k.$$

We sometimes refer to the user U_h's augmented verification vector $[1, \vec{v_h}] = (1, v_{h,1}, \ldots, v_{h,\omega})$ as $(v_{h,0}, \ldots, v_{h,\omega})$.

The polynomial F is determined by the $n(\omega+1)(\psi+1)$ unknown coefficients a_{ijk}. The coalition C has access to the following information:

1. The verification keys $\tilde{v}_1, \ldots, \tilde{v}_\omega$. We have, for $U_h \in C$,

   $$\tilde{v}_h(x, z) = F(x, \vec{y}, z)|_{\vec{y}=\vec{v_h}} = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{ijk} x^i v_{h,j} z^k$$

   Noting that $\tilde{v}_h$ is a polynomial with terms of the form $(c_{ik})_h x^i z^k$ for $0 \le i \le n-1$ and $0 \le k \le \psi$, we see that the coalition C has access to $n(\psi+1)(\omega)$ equations C_{ikh} in the unknowns a_{ijk}, where

   $$C_{ikh} \colon a_{i0k} + \sum_{j=1}^{\omega} a_{ijk} v_{h,j} = (c_{ik})_h$$

   for some (known) element $(c_{ik})_h \in \mathbb{F}_q$.
   We note that these equations

   $$\{C_{ikh} : 0 \le i \le n-1, 0 \le k \le \psi, 1 \le h \le \omega\} \tag{1}$$

   form a linearly independent set, as the rank of $\left\{(1, \vec{v_1}), \ldots, (1, \vec{v_\omega})\right\} \subseteq (\mathbb{F}_q)^{\omega+1}$ is ω. More details are provided in Appendix A.1.
2. The signing keys $s_1, \ldots, s_\omega$. We have, for $U_h \in C$,

   $$s_h(\vec{y}, z) = F(x, \vec{y}, z)|_{x=U_h} = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{ijk} U_h^i y_j z^k.$$

   Noting that s_h is a polynomial with terms of the form $(d_{jk})_h y_j z^k$, for $0 \le j \le \omega$ and $0 \le k \le \psi$, we have that C has access to $(\omega+1)(\psi+1)(\omega)$ equations D_{jkh} in the unknowns a_{ijk}, where

   $$D_{jkh} \colon \sum_{i=0}^{n-1} a_{ijk} U_h^i = (d_{jk})_h$$

   for some (known) element $(d_{jk})_h \in \mathbb{F}_q$.

Now, these equations, together with the equations from (1), are not a linearly independent set, due to the relationships between users' signing and verification keys. More specifically, for any users U_h and $U_{h'}$, we have

$$s_h(\vec{y}, z)|_{\vec{y}=v_{\vec{h'}}} = \tilde{v}_{h'}(x, z)|_{x=U_h}\,. \tag{2}$$

Equation (2) implies that for each $U_{h'} \in \mathcal{C}$ and each choice of $0 \leq k \leq \psi$, we have a set of ω relations among the $\omega + 1$ equations $\{D_{jkh} : 0 \leq j \leq \omega\}$.
Thus, the information gleaned from the coalition's signing information is contained in the set

$$\{D_{jkh} : 0 \leq k \leq \psi, 1 \leq h \leq \omega\}. \tag{3}$$

3. Up to ψ signatures $\sigma_{h,k'}$ from each user $U_h \notin C$, on messages $m_{h,k'}$ of C's choice, where $1 \leq k' \leq \psi$, with the exception that C can only access a signature $\sigma_{\zeta,k'}$ on a message $m_{\zeta,k'} \neq m$ with target signer U_ζ. Thus C has access to $n - \omega$ signatures of the form

$$\sigma_{h,k'}(\vec{y}) = s_h(\vec{y}, z)|_{z=m_{h,k'}} = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} a_{ijk} U_h^i y_j \left(m_{h,k'}\right)^k .$$

Note that $\sigma_{h,k'}$ is a polynomial with terms of the form $(b_j)_{h,k'}\, y_j$. Then C has access to $(\omega + 1)(\psi)(n - \omega)$ equations $B_{jhk'}$ in the unknowns a_{ijk}, where

$$B_{jhk'} \colon \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} a_{ijk} U_h^i (m_{h,k'})^k = (b_j)_{h,k'}$$

for some (known) element $(b_j)_{h,k'} \in \mathbb{F}_q$.
In a manner similar to the above analysis, we observe that

$$\sigma_{h,k'}(\vec{y})|_{\vec{y}=v_{\vec{h'}}} = \tilde{v}_{h'}(x, z)|_{x=U_h, z=m_{h,k'}} \tag{4}$$

for each $U_{h'} \in C$. Thus it suffices to consider the set

$$\{B_{0hk'} : 1 \leq k' \leq \psi\} \tag{5}$$

for each $U_h \notin C$.

4. Up to ψ_V query results from the oracle $\mathsf{Vrfy}_h^{\mathcal{O}}$ for $U_h \notin C$. In the following, we first consider the attack scenario without $\mathsf{Vrfy}^{\mathcal{O}}$ queries and then move to incorporate these queries into the analysis.

To summarize, the information obtained by the coalition C is contained in equation sets (1) and (3), together with, for each $U_h \notin C$, equation set (5). These equations form a linearly independent set; we provide the proof in Appendix A.1. We have a total of $n\omega\psi + n\omega + \omega + \psi n$ equations, which implies we have $n - \omega$ free variables in the given linear system.

With the given information, C can consider the polynomials $F'(x, \vec{y}, z)$ consistent with the known information about $F(x, \vec{y}, z)$. If a given polynomial F' is consistent with the known information about F, we say F' satisfies property $(*)$. We let

$$\mathcal{F} = \big\{F'(x, \vec{y}, z) : F' \text{ satisfies } (*)\big\}.$$

From above, we have $|\mathcal{F}| = q^{n-\omega}$.

Case 1: $U_\zeta \notin C$, $U_\nu \in C$

In this case, the goal of C is to produce a ζ-authentic signature; we wish to give an upper bound on C's probability of success, so we consider the most advantageous method by which C can create such a signature. If C creates a ζ-authentic signature (m, σ) consistent with C's known information, then this is equivalent to C finding U_ζ's signing key, $s_\zeta(\vec{y}, z)$. This follows because C would then have access to $\psi + 1$ points $\sigma(\vec{y})$, $\sigma_{\zeta,1}(\vec{y}), \ldots, \sigma_{\zeta,\psi}(\vec{y})$ on $s_\zeta(\vec{y}, z)$, which is a polynomial of degree ψ in z.

The above observation implies we can calculate the probability of success as

$$\frac{\big|\big\{F'(x, \vec{y}, z) \in \mathcal{F} : F'(U_\zeta, \vec{y}, z) = F(U_\zeta, \vec{y}, z)\big\}\big|}{\big|\big\{F'(x, \vec{y}, z) \in \mathcal{F}\big\}\big|}.$$

Using the same notation as before, if $F'(U_\zeta, \vec{y}, z) = F(U_\zeta, \vec{y}, z)$, we have the $\psi+1$ additional equations $\{D_{0k\zeta} : 0 \le k \le \psi\}$, rendering the equations $\{B_{0k'\zeta} : 1 \le k' \le \psi\}$ redundant. We can show the resulting set is linearly independent, so we have one additional restriction on F'. Recalling that we chose F' from a space of size $q^{n-\omega}$ initially, the coalition $\mathcal{C}$'s probability of success is

$$\frac{q^{n-\omega-1}}{q^{n-\omega}} = \frac{1}{q}.$$

Now, suppose C also has access to the $\mathsf{Vrfy}^{\mathcal{O}}$ oracle. We observe that if the query (m, σ) to $\mathsf{Vrfy}^{\mathcal{O}}_{h'}$ results in *True* (for some $U_{h'} \notin C$), and (m, σ) is consistent with C's information about F, then C has successfully determined U_ζ's signing key, $s_\zeta(\vec{y}, z)$. To see this, first note that if (m, σ) is consistent with C's information about F, then $\sigma(\vec{y}) = F'(x, \vec{y}, z)|_{x=U_\zeta, z=m}$ for some $F' \in \mathcal{F}$. This implies $F'(x, \vec{y}, z)|_{x=U_\zeta, z=m}$ agrees with $F(x, \vec{y}, z)|_{x=U_\zeta, z=m}$ on the $\omega + 1$ points $\vec{v_1}, \ldots, \vec{v_\omega}$, $\vec{v_{h'}}$. Since by assumption the augmented verification vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_\omega}]$, $[1, \vec{v_{h'}}] \in (\mathbb{F}_q)^{\omega+1}$ are linearly independent, we have

$$F'(x, \vec{y}, z)|_{x=U_\zeta, z=m} = F(x, \vec{y}, z)|_{x=U_\zeta, z=m}.$$

In other words, (m, σ) is a ζ-authentic signature. (This result is a consequence of basic linear algebra; we provide the relevant theory in Lemma A.2 of the Appendix.)

Now, any $F' \in \mathcal{F}$ also satisfies

$$F'(x, \vec{y}, z)|_{x=U_\zeta, z=m_{\zeta,k'}} = F(x, \vec{y}, z)|_{x=U_\zeta, z=m_{\zeta,k'}}$$

for $1 \leq k' \leq \psi$ and distinct messages $m_{\zeta,k'} \neq m$. That is, we have a total of $\psi + 1$ points at which $F'(x, \vec{y}, z)|_{x=U_\zeta}$ and $F(x, \vec{y}, z)|_{x=U_\zeta}$ agree as polynomials in z. Since F' and F are polynomials of degree ψ in z, this is sufficient to conclude

$$F'(x, \vec{y}, z)|_{x=U_\zeta} = F(x, \vec{y}, z)|_{x=U_\zeta} = s_\zeta(\vec{y}, z),$$

as desired. The probability of C finding s_ζ, however, is the probability of C choosing the correct F', which, as we show below, is $\frac{1}{q-\psi_V}$, where ψ_V is the number of queries to $\mathsf{Vrfy}^{\mathcal{O}}$ with result *False*.

We now consider ψ_V queries to $\mathsf{Vrfy}^{\mathcal{O}}$ with result *False*, supposing each query is consistent with C's view of the function F. We observe that each negative query eliminates (at most) one potential signing key for U_ζ. Given that the condition for success does not depend on the particular target verifier's verification key, $\vec{v_\nu}$, we can calculate the probability of success as before, this time allowing for information gleaned from the ψ_V negative queries. We write $\bar{s_\zeta}^1, \ldots, \bar{s_\zeta}^{\psi_V}$ for these eliminated signing keys, and for readability, we write $F'_\zeta(\vec{y}, z)$ for $F'(x, \vec{y}, z)|_{x=U_\zeta}$.

We first need to calculate $\left|\left\{F'(x, \vec{y}, z) \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right|$, i.e., the number of possible functions F' consistent with C's view of F. We have

$$\begin{aligned}&\left|\left\{F' \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right| \\ &\quad = \left|\left\{F' \in \mathcal{F}\right\}\right| - \left|\left\{F' \in \mathcal{F} : F'_\zeta \in \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right|.\end{aligned}$$

We assume the events $F'_\zeta = \bar{s_\zeta}^1, \ldots, F'_\zeta = \bar{s_\zeta}^{\psi_V}$ are disjoint, since if $\bar{s_\zeta}^i = \bar{s_\zeta}^j$ for some $1 \leq i, j \leq \psi_V$, this is equivalent to fewer verification oracle queries. Following the same reasoning as before, we then have

$$\begin{aligned}\left|\left\{F' \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right| &= \left|\left\{F' \in \mathcal{F}\right\}\right| - \sum_{i=1}^{\psi_V} \left|\{F' \in \mathcal{F} : F'_\zeta = \bar{s_\zeta}^i\}\right| \\ &= q^{n-\omega} - \psi_V q^{n-\omega-1} \\ &= q^{n-\omega-1}(q - \psi_V).\end{aligned}$$

We therefore calculate C's probability of success as:

$$\begin{aligned}&\frac{\left|\left\{F'(x, \vec{y}, z) \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}, F'_\zeta = s_\zeta\right\}\right|}{\left|\left\{F'(x, \vec{y}, z) \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right|} \\ &\quad = \frac{\left|\left\{F'(x, \vec{y}, z) \in \mathcal{F} : F'_\zeta = s_\zeta\right\}\right|}{\left|\left\{F'(x, \vec{y}, z) \in \mathcal{F} : F'_\zeta \notin \{\bar{s_\zeta}^1, \bar{s_\zeta}^2, \ldots, \bar{s_\zeta}^{\psi_V}\}\right\}\right|} \\ &\quad = \frac{q^{n-\omega-1}}{q^{n-\omega-1}(q - \psi_V)} = \frac{1}{q - \psi_V}.\end{aligned}$$

Case 2: $U_\zeta \notin C$, $U_\nu \notin C$

Now suppose $U_\nu \notin C$. Ostensibly the goal of C is to produce a (ζ,ν)-acceptable signature. Note that in order for a signature pair (m,σ) with claimed signer U_ζ to pass U_ν's verification check, (m,σ) must satisfy $\sigma(\vec{y})|_{\vec{y}=\vec{v_\nu}} = \tilde{v}_\nu(x,z)|_{x=U_\zeta, z=m}$. In particular, if (m,σ) is consistent with both U_ν's verification key and with F, then the same analysis as in the previous case implies that (m,σ) is a ζ-authentic signature, and indeed that C has determined s_ζ. Thus, the set of known information $(*)$ does not help create a (ζ,ν)-fraudulent signature. For the case of creating a (ζ,ν)-fraudulent signature, the most powerful collusion C includes the signer U_ζ, which we consider next.

Case 3: $U_\zeta \in C$, $U_\nu \notin C$

Here C's goal is to produce a (ζ,ν)-fraudulent signature. Since the polynomial F and the set of verification vectors $\vec{v_1},\ldots,\vec{v_n} \in (\mathbb{F}_q)^\omega$ are chosen independently, we see that the signing keys of C and sample signatures from $U_h \notin C$ have no bearing on the probability distribution for the key $\vec{v_\nu}$.

Recall that for any subset of the set $\{\vec{v_1},\ldots,\vec{v_n}\}$ of size $\omega+1$, the corresponding subset of size $\omega+1$ formed from the new vectors $[1,\vec{v_1}],\ldots,[1,\vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. Therefore, knowledge of the keys $\vec{v_h}$ for $U_h \in C$ does affect the probability distribution for the key $\vec{v_\nu}$. In particular, C is aware that $[1,\vec{v_\nu}] \neq \sum_{j=1}^{\omega} k_j[1,\vec{v_j}]$ for any choice of $\{k_1,\ldots,k_\omega \in \mathbb{F}_q : \sum_{j=1}^{\omega} k_j = 1\}$. That is, given $\vec{v_1},\ldots,\vec{v_\omega}$, there are $q^\omega - q^{\omega-1}$ choices for $\vec{v_\nu}$, any of which are equally likely. We write V for the set of possible vectors $\vec{v_\nu}$.

Now suppose we want to create a (ζ,ν)-fraudulent signature $\sigma'(\vec{y})$ on a message m. Suppose $\sigma(\vec{y}) = b_0 + \sum_{j=1}^{\omega} b_j y_j$ is the ζ-authentic signature on m. Then writing

$$\sigma'(\vec{y}) = b'_0 + \sum_{j=1}^{\omega} b'_j y_j,$$

we need $\sigma(\vec{v_\nu}) = \sigma'(\vec{v_\nu})$, but $(b_0,\ldots,b_\omega) \neq (b'_0,\ldots,b'_\omega)$.

In other words, C needs to find a nonzero vector $\vec{\beta} = (b_0 - b'_0,\ldots,b_\omega - b'_\omega)$ satisfying $\vec{\beta}\cdot[1,\vec{v_\nu}] = 0$. The probability of success is then calculated as

$$\begin{aligned}\max_{\vec{\beta}} \frac{\left|\{\vec{v_\nu} \in V : \vec{\beta}\cdot[1,\vec{v_\nu}] = 0\}\right|}{\left|\{\vec{v_\nu} \in V\}\right|} &\leq \max_{\vec{\beta}} \frac{\left|\{\vec{v_\nu} \in (\mathbb{F}_q)^\omega : \vec{\beta}\cdot[1,\vec{v_\nu}] = 0\}\right|}{\left|\{\vec{v_\nu} \in V\}\right|} \\ &= \frac{q^{\omega-1}}{q^\omega - q^{\omega-1}} = \frac{1}{q-1}.\end{aligned}$$

We now consider $\mathsf{Vrfy}^{\mathcal{O}}$ queries. We observe that a positive $\mathsf{Vrfy}_\nu^{\mathcal{O}}$ query (m,σ) allows the coalition C to win the game Sig-forge$_{C,\Pi}(k)$, so we consider the probability of success given ψ_V negative $\mathsf{Vrfy}_\nu^{\mathcal{O}}$ queries, since this gives the best chance of success. (Note that it is

also possible, albeit extremely unlikely, that a positive $\mathsf{Vrfy}_\nu^{\mathcal{O}}$ query here results in a forgery that is ζ-authentic. The coalition C also wins in this instance, but we are not concerned with ζ-authentic forgeries here, as the best approach to producing these types of forgeries is analyzed in Case 1.)

We let V' be the set of possible vectors $\vec{v_\nu}$ given the new knowledge gleaned from the ψ_V negative query vectors $\vec{\beta_1}, \ldots, \vec{\beta_{\psi_V}}$. That is,

$$V' = \left\{\vec{v_\nu} \in V : \vec{\beta_1} \cdot [1, \vec{v_\nu}] \neq 0, \ldots, \vec{\beta_{\psi_V}} \cdot [1, \vec{v_\nu}] \neq 0\right\}.$$

Now,

$$\begin{aligned}
\left|\left\{\vec{v_\nu} \in V'\right\}\right| &= \left|\left\{\vec{v_\nu} \in V\right\}\right| - \left|\left\{\vec{v_\nu} \in V : \vec{\beta_1} \cdot [1, \vec{v_\nu}] = 0 \text{ or } \cdots \text{ or } \vec{\beta_{\psi_V}} \cdot [1, \vec{v_\nu}] = 0\right\}\right| \\
&\geq \left|\left\{\vec{v_\nu} \in V\right\}\right| - \left|\left\{\vec{v_\nu} \in (\mathbb{F}_q)^\omega : \vec{\beta_1} \cdot [1, \vec{v_\nu}] = 0 \text{ or } \cdots \text{ or } \vec{\beta_{\psi_V}} \cdot [1, \vec{v_\nu}] = 0\right\}\right| \\
&\geq \left|\left\{\vec{v_\nu} \in V\right\}\right| - \sum_{i=1}^{\psi_V} \left|\left\{\vec{v_\nu} \in (\mathbb{F}_q)^\omega : \vec{\beta_i} \cdot [1, \vec{v_\nu}] = 0\right\}\right| \\
&= (q^\omega - q^{\omega-1}) - \psi_V q^{\omega-1} \\
&= q^{\omega-1}(q - \psi_V - 1).
\end{aligned}$$

The probability of success is then calculated as

$$\begin{aligned}
\max_{\vec{\beta}} \frac{\left|\left\{\vec{v_\nu} \in V' : \vec{\beta} \cdot [1, \vec{v_\nu}] = 0\right\}\right|}{\left|\left\{\vec{v_\nu} \in V'\right\}\right|} &\leq \max_{\vec{\beta}} \frac{\left|\left\{\vec{v_\nu} \in (\mathbb{F}_q)^\omega : \vec{\beta} \cdot [1, \vec{v_\nu}] = 0\right\}\right|}{\left|\left\{\vec{v_\nu} \in V'\right\}\right|} \\
&\leq \frac{q^{\omega-1}}{q^{\omega-1}(q - \psi_V - 1)} = \frac{1}{q - \psi_V - 1}.
\end{aligned}$$

This completes the proof. □

Remark 2.11. The linear independence assumption in the above construction is not necessary, as observed by Hanaoka et al. [38], but it does simplify the security analysis. If the linear independence assumption is not satisfied, we must take into account the rank of $\{\vec{v_h} : U_h \in C\}$, which may be strictly less than ω. In this case, the coalition C has less information, but the proof is similar. We can also increase the robustness of the construction against verification oracle queries by using a polynomial $F(x, y_1, \ldots, y_{\omega+\tau}, z)$ of the same form as above, where $\tau > 0$. This achieves security as outlined in Theorem 2.11, where the coalition has, in addition, achieved up to τ successful verification oracle queries. This technique is used by Shikata et al. [65] in their construction, although it is not explained.

2.8. USS Schemes with Key Insulation

Key exposure is a major concern in any cryptosystem. In traditional public-key cryptography, Dodis et al. [23] introduced the notion of *key insulation*, in which a user's secret information is split between a physically secure (and perhaps computationally limited)

device, H, and an insecure device with temporary secret keys that are refreshed at intervals with information sent by H. These notions have been applied to signatures in the traditional setting by Dodis et al. [24] and to unconditionally secure multi-receiver authentication codes and key agreement by Seito et al. [63] and Seito and Shikata [62]. In this section, we concern ourselves with *key exposure of a user's signing information*. Our main goal is to provide an example of how our basic USS security model might be extended to incorporate more complicated security notions, such as key insulation. Our definitions are extensions of those provided by Seito et al. [63] and Seito and Shikata [62] to the signature setting, keeping in mind the original goals of Dodis et al. [24].

The basic idea is as follows. A user's signing information is split into a "master" signing key stored on a secure device, temporary secret-signing keys, which are derived from an initial secret (stored on an insecure device), and key-updating information (which is sent at intervals from the secure device). For each signer, we want the scheme to be robust against exposure of *either* that user's master signing key *or* some (strict) subset of the user's temporary signing keys, *but not both*. The overall scheme should be secure provided these exposure criteria hold for all honest users; this property is called *strong key insulation*.

We begin by providing a formal definition of an unconditionally secure signature scheme with key insulation (KI-USS) in Section 2.8.1. In Section 2.8.2, we give an extension of our basic security model from Section 2.4 to the key-insulation setting. Once we have established our formal security notions and model, we give an extension to the USS construction from Hanaoka et al. [38] (which we analyzed in Section 2.7) in Section 2.9. The extension is inspired by a multi-receiver authentication code construction presented by Seito et al. [63].

2.8.1. Preliminary definitions.

We require the following definitions.

Definition 2.19. An *unconditionally secure signature scheme with key insulation* (or *KI-USS scheme*) Π consists of a trusted initializer TI, a set $\mathcal{U}$ of n users, a set $\mathcal{H}$ of n secure devices, and a tuple of seven spaces $(\mathcal{T}, \mathcal{X}, \Sigma, \mathcal{V}, \mathcal{I}, \mathcal{MK}, \mathcal{SK})$, together with algorithms Gen and $\left\{\mathsf{Sign}_\zeta, \mathsf{Vrfy}_\zeta, \mathsf{MKUpd}_\zeta, \mathsf{SKUpd}_\zeta\right\}_{1\leq\zeta\leq n}$, satisfying the following:

- The set $\mathcal{U} = \{U_1, \ldots U_n\}$ consists of n possible users.
- $\mathcal{H} = \{H_1, \ldots, H_n\}$ is a set of n secure devices, where each $H_i \in \mathcal{H}$ is the secure device for user $U_i \in \mathcal{U}$.
- $\mathcal{T} = \{0, 1, 2, \ldots, N\}$ is a set of time periods.
- $\mathcal{X}$ is a finite set of possible messages.
- Σ is a finite set of possible signatures.
- $\mathcal{V}$ is a finite set of possible (secret) verification information.
- $\mathcal{I}$ is a finite set of possible secret-key-updating information (to keep track of time periods).
- $\mathcal{MK}$ is a finite set of possible (secret) master keys.
- $\mathcal{SK}$ is a finite set of possible secret signing keys. The set $\mathcal{SK}^t$ is the set of possible signing keys at time period t.

- The *key-generation algorithm* Gen takes as input 1^k, where k is a security parameter, and the total number of time periods N and outputs a master secret key $mk^* := (mk_1, \ldots, mk_n) \in \mathcal{MK}^n$ and initial signing key information $sk^* := \left(sk_1^0, \ldots, sk_n^0\right) \in \mathcal{SK}^n$, together with verification keys $\{v_\zeta \in \mathcal{V} : 1 \leq \zeta \leq n\}$.
- For each $U_\zeta \in \mathcal{U}$, the *master-key-updating algorithm* $\mathsf{MKUpd}_\zeta \colon \mathcal{MK} \times \mathcal{T} \to \mathcal{I}$ for user U_ζ takes as input U_ζ's master key mk_ζ, and a time period $t \in T$, and returns secret key-updating information $mk_\zeta^{(t-1,t)} \in \mathcal{I}$. The key-updating information $mk_\zeta^{(t-1,t)} \in \mathcal{I}$ is used by U_ζ in order to update his signing key from time period $t-1$ to time period t, as described by the next algorithm SKUpd_ζ.
- For each $U_\zeta \in \mathcal{U}$, the *signing-key-updating algorithm* $\mathsf{SKUpd}_\zeta \colon \mathcal{T} \times \mathcal{SK} \times \mathcal{I} \to \mathcal{SK}$ takes as input a time period $t \in \mathcal{T}$, a secret signing key $sk_\zeta^{(t-1)}$ for time period $t-1$, and secret key-updating information $mk_\zeta^{(t-1,t)}$, and returns a signing key $sk_\zeta^t \in \mathcal{SK}^t$ for time period t.
- For each $U_\zeta \in \mathcal{U}$, the *signing algorithm* $\mathsf{Sign}_\zeta \colon \mathcal{T} \times \mathcal{X} \times \mathcal{SK} \to \Sigma$ takes as input a time period $t \in \mathcal{T}$ satisfying $t > 0$, a message $x \in \mathcal{X}$, and a signing key $sk_\zeta^t \in \mathcal{SK}$, and returns a signature $\sigma \in \Sigma$. We let Sign_ζ^t denote the algorithm $\mathsf{Sign}_\zeta\left(t, \cdot, sk_\zeta^t\right)$.
- For each $U_\zeta \in \mathcal{U}$, the *verification algorithm* $\mathsf{Vrfy}_\zeta \colon \mathcal{X} \times \mathcal{T} \times \Sigma \times \mathcal{U} \times \mathcal{V} \to \{True, False\}$ takes as input a message $x \in \mathcal{X}$, a time period $t \in \mathcal{T}$, a signature $\sigma \in \Sigma$, a signer $U_\nu \in \mathcal{U}$, and verification key $v_\zeta \in \mathcal{V}$, and outputs either *True* or *False*. For each user U_ζ, we let Vrfy_ζ^t denote the algorithm $\mathsf{Vrfy}_\zeta\left(\cdot, t, \cdot, \cdot, v_\zeta\right)$.

Scheme Phases:

1. *Key Generation phase.* The TI runs Gen and securely distributes $\left(mk_\zeta, sk_\zeta^0, v_\zeta\right)$ to the corresponding user U_ζ for all $U_\zeta \in \mathcal{U}$. The TI then deletes all keys from his memory. The user U_ζ places his master key mk_ζ on his secure device H_ζ and then deletes mk_ζ from his memory.
2. *Update phase.* To update signing information for a user U_ζ from time period $t-1$ to period t, the secure device H_ζ runs $\mathsf{MKUpd}_\zeta(mk_\zeta, t)$ and sends the output $mk_\zeta^{(t-1,t)}$ to U_ζ via a secure channel. The user U_ζ then runs $\mathsf{SKUpd}(t, sk_\zeta^{(t-1)}, mk^{(t-1,t)})$, which outputs sk_ζ^t, the signing key for the new time period t. The user U_ζ then deletes $sk_\zeta^{(t-1)}$ and $mk^{(t-1,t)}$ from his memory.
3. *Signing phase.* To sign a message $x \in \mathcal{X}$ during a time period t, a user U_ζ runs his signing algorithm $\mathsf{Sign}_\zeta^t(x)$, which outputs a signature $\sigma \in \Sigma$. The user U_ζ then forms the signature triple (x, t, σ).
4. *Verification phase.* To verify a signature triple (x, t, σ) from a signer U_ν, a user U_ζ runs his verification algorithm $\mathsf{Vrfy}_\zeta^t\left(x, \sigma, U_\nu\right)$. If the output of $\mathsf{Vrfy}_\zeta^t\left(x, \sigma, U_\nu\right)$ is *True*, then U_ζ believes that the signature pair was actually produced by U_ν's signing algorithm during time period t as claimed.

It is required that, for every k, for every N, for every set $\left\{\mathsf{Sign}_\zeta, \mathsf{Vrfy}_\nu : 1 \leq \zeta, \nu \leq n\right\}$ output by $\mathsf{Gen}\left(1^k, N\right)$, for every pair U_ζ, $U_\nu \in \mathcal{U}$, and for every $x \in \mathcal{X}$ and $t \in \mathcal{T}$ such that $t > 0$, it holds that

$$\mathsf{Vrfy}_\nu^t\left(x, \mathsf{Sign}_\zeta^t(x), U_\zeta\right) = \mathit{True}.$$

REMARK 2.12. We are treating *deterministic* signature schemes only, in the sense that all algorithms except Gen are deterministic, although the above definition can easily be extended to the randomized setting.

2.8.2. Security model. The concepts of authentic, acceptable, and fraudulent signatures are defined as before.

Definition 2.20. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ during a time period $t \in \mathcal{T}$ is ζ-*authentic* if $\sigma = \mathsf{Sign}_\zeta^t(x)$.

Definition 2.21. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ during a time period $t \in \mathcal{T}$ is (ζ, ν)-*acceptable* if $\mathsf{Vrfy}_\nu^t\left(x, \sigma, U_\zeta\right) = \mathit{True}$.

Definition 2.22. A signature $\sigma \in \Sigma$ on a message $x \in \mathcal{X}$ during a time period $t \in \mathcal{T}$ is (ζ, ν)-*fraudulent* if σ is (ζ, ν)-acceptable but not ζ-authentic.

Informally, we wish to guard against two types of possible key exposure for each honest user U_ℓ. We want the scheme to be secure against *either* (but not both) of the following attacks on honest users $U_\ell \in \mathcal{U}$:

- *Signing key exposure*: Compromise of user U_ℓ's signing keys from the insecure device for up to γ time periods
- *Master key exposure*: Compromise of U_ℓ's secure device, where mk_ℓ is stored.

We need to define the following oracles. The first two of these oracles are used to model possible key exposure for honest users, and the latter two are direct generalizations of the signing and verification oracles used for regular USS schemes.

- The SigningExposure$^{\mathcal{O}}(\cdot, \cdot)$ *oracle*; this oracle takes as input a user $U_\ell \in \mathcal{U}$ and a time period $t \in \mathcal{T}$ (where $t > 0$) and outputs U_ℓ's signing information for period t, namely $\mathsf{Sign}_\ell^t(\cdot)$. This oracle is used to model compromise of U_ℓ's insecure device for up to γ time periods, where U_ℓ's temporary signing keys are stored.
- The MasterExposure$^{\mathcal{O}}(\cdot)$ oracle; this oracle takes as as input a user $U_\ell \in \mathcal{U}$ and outputs U_ℓ's master key mk_ℓ. This oracle is used to model compromise of U_ℓ's secure device, where the master key mk_ℓ is stored.
- The $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot, \cdot)$ *oracle*; this oracle takes as input a message $x \in \mathcal{X}$ and time period $t \in \mathcal{T}$ and outputs an ℓ-authentic signature on the message x for time t.
- The $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot, \cdot, \cdot, \cdot)$ *oracle*; this oracle takes as input a signature triple (x, t, σ) (i.e., a message $x \in \mathcal{X}$, a time period $t \in \mathcal{T}$, and a signature $\sigma \in \Sigma$) and a signer U_ζ, and runs user U_ℓ's verification algorithm on input (x, t, σ, U_ζ), outputting *True* or *False*.

We now define the formal model as follows:

Definition 2.23. Let Π be a KI-USS scheme (with notation as defined in Definition 2.19), with security parameter k. Let $C \subseteq \mathcal{U}$ be a coalition of at most ω users and let ψ_S, ψ_V, and γ be positive integers. We define the following *signature game* KI-Sig-forge$_{C,\Pi}(k)$ with target signer U_ζ and verifier U_ν:

1. $\mathsf{Gen}\left(1^k\right)$ is run to obtain the pair (mk^*, sk^*).
2. The coalition C is given bounded access to the following oracles: MasterExposure $\mathcal{O}(\cdot)$, SigningExposure $\mathcal{O}(\cdot,\cdot)$, $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot,\cdot)$, and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,\cdot,\cdot)$. The rules for oracle access are as follows:
   (a) For each $U_\ell \notin C$, the coalition C is permitted *only one* of the following:
      - SigningExposure$^{\mathcal{O}}(U_\ell, t)$ for up to γ time periods $t \in \mathcal{T}\backslash\{0\}$;
      - MasterExposure$^{\mathcal{O}}(U_\ell)$.

   We let $T' = \left\{t \in \mathcal{T}\colon C \text{ has accessed SigningExposure}^{\mathcal{O}}(U_\zeta, t)\right\}$.
   (b) The coalition C is given bounded access to the $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot,\cdot)$ and $\mathsf{Vrfy}_\ell^{\mathcal{O}}(\cdot,\cdot,\cdot,U_\zeta)$ oracles for ℓ satisfying $U_\ell \notin C$. In particular, C is allowed a total of ψ_S and ψ_V queries to the $\mathsf{Sign}^{\mathcal{O}}$ and $\mathsf{Vrfy}^{\mathcal{O}}$ oracles, respectively, with at most $\psi_S/(n - |C|)$ queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot)$ for each ℓ satisfying $U_\ell \notin C$. It should be noted that C has unlimited access to the signing and verification algorithms of any $U_\ell \in C$. For each time period $t \in \mathcal{T}$, we let $\mathcal{Q}_t$ denote the set of messages that the coalition submitted as queries to the $\mathsf{Sign}_\zeta^{\mathcal{O}}(\cdot, t)$ oracle. Note that $\mathcal{Q}_t$ does not contain messages submitted as queries to $\mathsf{Sign}_\ell^{\mathcal{O}}(\cdot, t)$ for $\ell \neq \zeta$.
3. The coalition C outputs a signature triple (x, t, σ).
4. The output of the game is defined to be 1 if and only if one of the following conditions is met:
   (a) $U_\nu \notin C$ and σ is a (ζ, ν)-fraudulent signature on x for period t; or
   (b) $U_\zeta \notin C$ and σ is a ζ-authentic signature on x for period t, where $x \notin \mathcal{Q}_t$ and $t \notin T'$.

Definition 2.24. Let Π be a KI-USS scheme (with notation as defined in Definition 2.19) with security parameter k and let $\epsilon(k)$ be a negligible function of k. We say Π is *strongly* $(\omega, \gamma, \psi_S, \psi_V, \epsilon)$-*unforgeable* if for all coalitions C of at most ω users, and all choices of target signer U_ζ and verifier U_ν, it holds that

$$\Pr\left[\text{KI-Sig-forge}_{C,\Pi}(k) = 1\right] \leq \epsilon(k).$$

Given the nature of signing key exposure, it is reasonable to consider a scenario in which two or more *consecutive* time periods are compromised by the adversary. In this case, it is quite possible (or even likely) that the adversary gains access not only to the signing keys from the exposed periods, but also the key-updating information sent from the user's secure device between those compromised time periods. To protect against this, it is useful to consider the notion of *secure key updates* [**23**], which says that the combination

of signing information from two consecutive exposed periods $t-1$ and t, together with the key-updating information between these periods, should be equivalent to the signing information from these two periods alone.

Definition 2.25. Let Π be a strongly $(\omega, \gamma, \psi_S, \psi_V, \epsilon)$-unforgeable KI-USS scheme. Suppose a coalition C of at most ω users plays the signature game of Definition 2.23, with the modification that any time C accesses the oracles $\text{SigningExposure}^{\mathcal{O}}(U_\ell, t-1)$ and $\text{SigningExposure}^{\mathcal{O}}(U_\ell, t)$ for a user $U_\ell \notin C$ and two consecutive time periods $t-1$ and t, the coalition C receives the additional information $mk_\ell^{(t-1,t)}$, together with U_ℓ's signing information from periods $t-1$ and t. If Π satisfies Definition 2.24 with this new signature game, we say that Π has *secure key updates*.

2.9. Construction: USS Scheme with Key Insulation

We now give an extension to the USS construction from Hanaoka et al. [**38**], which we analyzed in Section 2.7. The extension presented here uses ideas from a multi-receiver authentication code construction presented by Seito et al. [**63**].

The construction given here is very similar to the basic construction given in Section 2.7, except that we need our polynomial construction to be divided into two pieces, so that we can split each user's signing algorithm into an initial signing key which is stored on the user's insecure device and a master signing key which is stored on the user's secure device.

To that end, we use a polynomial F of the same form as the basic construction and a polynomial mk, which has a similar form as F but is extended to take into account time periods. The polynomials F and mk are used to construct (by substituting a user's identity into these polynomials, as before) each user's initial signing key and master signing key, respectively. A user's overall signing information for a particular time period is the sum of these two polynomials evaluated at the user's identity and the given time period.

2.9.1. Key pair generation. Let q be a prime power such that $q > n$. (In practice, we pick q to be much larger than n.) Let $\mathbb{F}_q$ be a finite field with q elements. The TI picks n *verification vectors* $\vec{v_1}, \ldots, \vec{v_n} \in (\mathbb{F}_q)^\omega$ uniformly at random for users $U_1, \ldots U_n$, respectively, subject to one additional constraint. For technical reasons, we assume the verification vectors $\vec{v_1}, \ldots \vec{v_n} \in (\mathbb{F}_q)^\omega$ satisfy the additional property that for any subset of size $\omega+1$, the corresponding subset of size $\omega+1$ formed from the new vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. We assume user identities $U_1, \ldots, U_n$ and time periods $\{1, \ldots, N\}$ have a representation as elements in $\mathbb{F}_q$ in some suitable (and public) way.

The TI constructs two polynomials:

1. The polynomial $F(x, y_1, \ldots, y_\omega, z)$, where
$$F(x, y_1, \ldots, y_\omega, z) = \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} a_{i0k0} x^i z^k + \sum_{i=0}^{n-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} a_{ijk0} x^i y_j z^k,$$
where the coefficients $a_{ijk0} \in \mathbb{F}_q$ are chosen uniformly at random.
2. The polynomial $mk(x, y_1, \ldots, y_\omega, z, t)$, where
$$mk(x, y_1, \ldots, y_\omega, z, t) = \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} \sum_{\ell=1}^{\gamma} a_{i0k\ell} x^i z^k t^\ell + \sum_{i=0}^{n-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=1}^{\gamma} a_{ijk\ell} x^i y_j z^k t^\ell$$

For each user U_ζ for $1 \leq \zeta \leq n$, the TI computes the *initial signing key*
$$sk_\zeta^0(y_1, \ldots, y_\omega, z) = F(x, y_1, \ldots, y_\omega, z)|_{x=U_\zeta},$$
the *master signing key*
$$mk_\zeta(y_1, \ldots, y_\omega, z, t) = mk(x, y_1, \ldots, y_\omega, z, t)|_{x=U_\zeta},$$
and the *verification key*
$$\tilde{v}_\zeta(x, z, t) = F(x, y_1, \ldots, y_\omega, z)|_{(y_1,\ldots,y_\omega)=\vec{v_\zeta}} + mk(x, y_1, \ldots, y_\omega, z, t)|_{(y_1,\ldots,y_\omega)=\vec{v_\zeta}}.$$
It is assumed the TI sends sk_ζ^0, mk_ζ, $\vec{v_\zeta}$, and $\tilde{v}_\zeta$ to the corresponding user via a secure channel and deletes the information from his memory afterwards. The user U_ζ places his master signing key $mk_\zeta(y_1, \ldots, y_\omega, z, t)$ on his secure device H_ζ and deletes this information from his memory.

2.9.2. Updating phase. To update his signing key from a time period t_0 to the next time period t_1, a user U_ζ requests key-updating information from the secure device H_ζ. The device H_ζ computes
$$mk_\zeta^{(t_0,t_1)}(y_1, \ldots, y_\omega, z) := mk_\zeta(y_1, \ldots, y_\omega, z, t)|_{t=t_1} - mk_\zeta(y_1, \ldots, y_\omega, z, t)|_{t=t_0}$$
and sends this polynomial via a secure channel to U_ζ.

The user U_ζ then computes
$$\mathsf{Sign}_\zeta^{(t_1)}(y_1, \ldots, y_\omega, z) = \mathsf{Sign}_\zeta^{(t_0)}(y_1, \ldots, y_\omega, z) + mk_\zeta^{(t_0,t_1)}(y_1, \ldots, y_\omega, z),$$
where the signing key for time period $t = 1$ is defined by
$$\mathsf{Sign}_\zeta^{(1)} = sk_\zeta^0(y_1, \ldots, y_\omega, z) + mk_\zeta^{(0,1)}(x, y_1, \ldots, y_\omega, z).$$

REMARK 2.13. For a given time period $t_0 > 0$, user U_ζ's signing key is as follows:
$$\mathsf{Sign}_\zeta^{(t_0)}(y_1, \ldots, y_\omega, z) = F(x, y_1, \ldots, y_\omega, z)|_{x=U_\zeta} + mk(x, y_1, \ldots, y_\omega, z, t)|_{x=U_\zeta, t=t_0}.$$

2.9.3. Signature generation and verification. For a message $m \in \mathbb{F}_q$ during time period t_0, U_ζ generates a signature by

$$\sigma(y_1, \ldots, y_\omega) = \mathsf{Sign}_\zeta^{(t_0)}(y_1, \ldots, y_\omega, z)\big|_{z=m}.$$

To verify a signature pair (t_0, σ) from U_ζ on a message m, a user U_ν checks that

$$\sigma(y_1, \ldots, y_\omega)|_{(y_1,\ldots,y_\omega)=\vec{v_\nu}} = \tilde{v}_\nu(x, z, t)|_{x=U_\zeta, z=m, t=t_0}\,.$$

REMARK 2.14. As in the basic construction, the parameter ω determines the maximum number of colluders the scheme protects against and the parameter ψ determines the maximum number of signatures (on unique messages) each user can produce without revealing their signing information. Similarly, the parameter γ is the maximum number of time periods for which a user U_h's temporary signing key can be compromised (so long as U_h's master signing key is not exposed).

2.9.4. Security analysis. We consider the security of this construction in a restricted model, specified as follows. We let $\mathcal{Q}$ denote the set of messages that the coalition submitted as queries to the $\mathsf{Sign}_\zeta^{\mathcal{O}}$ oracle. We then replace Condition 4(b) of Definition 2.23 with the following:

4(b) $U_\zeta \notin C$ and σ is a ζ-authentic signature on x for period t, where $x \notin \mathcal{Q}$ and $t \notin T'$.

This weakened condition allows forgeries (x, t, σ) for signer U_ζ in the case where a ζ-authentic signature for some time $t' \neq t$ is known (and U_ζ's signing key for time period t has not been exposed). We can mitigate the impact of this type of forgery in our construction by assuming messages m contain effective dates for signatures. In this sense, an adversary can create a new signature on m for a different time period once he has seen the first signature on m, but the effective date of the signature will remain the same, i.e., this type of forgery will be detectable.

Given q, we define the security parameter to be k, where $k = \log_2 q$. We consider the game KI-Sig-forge$_{C,\Pi}(k)$ and calculate the probability that the output is 1. In particular, we consider the probability that the coalition C produces a signature triple (m, t', σ) satisfying Conditions 4(a) and 4(b) of Definition 2.23 separately. Here C is allowed, for each $U_h \notin C$, either γ queries to SigningExposure $\mathcal{O}(U_h, \cdot)$ or the single query MasterExposure $\mathcal{O}(U_h)$, but not both. We prove the scheme is unforgeable with respect to coalitions C of size at most ω, where C is allowed $\psi_S = (n - \omega)\psi$ oracle queries to $\mathsf{Sign}^{\mathcal{O}}$ (where ψ is the total number of $\mathsf{Sign}_h^{\mathcal{O}}$ oracle queries allowed for each user $U_h \notin C$), and where the number of $\mathsf{Vrfy}^{\mathcal{O}}$ queries, say ψ_V, is arbitrary. (As shown in the following theorem, the probability that C creates a successful forgery depends on this value ψ_V.) That is, we allow C to have at most ω members and to have access to ψ sample signatures from each user $U_h \notin C$. (This is consistent with the fact that in this USS scheme, each user is allow to produce at most ψ signatures, so the bound on oracle access to $\mathsf{Sign}_h^{\mathcal{O}}$ for each user $U_h \notin C$ must be ψ.)

Theorem 2.12. *Under the above assumptions, C outputs a signature triple (m, t', σ) in the game* KI-Sig-forge$_{C,\Pi}(k)$ *of Definition 2.23 satisfying Condition 4(a) with probability at most $\frac{1}{q-\psi_V-1}$ and Condition 4(b) with probability at most $\frac{1}{q-\psi_V}$.*

Proof. Recall the assumption that the verification vectors $\vec{v_1}, \ldots, \vec{v_n} \in (\mathbb{F}_q)^\omega$ satisfy the additional property that for any subset of size $\omega+1$, the corresponding subset of size $\omega+1$ formed from the new vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. We use this fact throughout the proof.

We consider the strongest possible coalition C. To this end, we consider a coalition of size ω whose verification vectors form a linearly independent set. Lemma A.1 implies that this is always possible. Without loss of generality, assume our adversaries are $C = \{U_1, \ldots, U_\omega\}$, with target signer U_ζ and target verifier U_ν. We assume that the coalition C outputs a signature (t', σ) for some choice of time period t' and message m, with claimed signer U_ζ.

For ease of notation, define $y_0 = 1$ and let $\vec{y}$ denote the vector $(y_0, y_1, \ldots, y_\omega)$. Let $G(x, \vec{y}, z, t)$ denote $F(x, \vec{y}, z) + mk(x, \vec{y}, z, t)$. Then

$$G(x, \vec{y}, z, t) = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=0}^{\gamma} a_{ijk\ell} x^i y_j z^k t^\ell.$$

We sometimes refer to a user U_h's augmented verification vector $[1, \vec{v_h}]$ as $(v_{h,0}, \ldots, v_{h,\omega})$.

The polynomial F is determined by the $n(\omega+1)(\psi+1)(\gamma+1)$ unknown coefficients $a_{ijk\ell}$. The coalition C has access to the following information:

1. The verification keys $\tilde{v}_1, \ldots, \tilde{v}_\omega$. We have, for $U_h \in C$,

$$\tilde{v}_h(x, z, t) = G(x, \vec{y}, z, t)|_{\vec{y}=\vec{v_h}} = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=0}^{\gamma} v_{h,j} a_{ijk\ell} x^i z^k t^\ell.$$

   Noting that $\tilde{v}_h$ is a polynomial with terms of the form $(c_{ik\ell})_h x^i z^k t^\ell$ for $0 \le i \le n-1$, $0 \le k \le \psi$, and $0 \le \ell \le \gamma$, we have that C has access to $n(\psi+1)(\gamma+1)$ equations $C_{ik\ell h}$ in the unknown coefficients $a_{ijk\ell}$ for each $U_h \in C$, where

$$C_{ik\ell h} \colon \sum_{j=0}^{\omega} v_{h,j} a_{ijk\ell} = (c_{ik\ell})_h$$

   for some (known) $(c_{ik\ell})_h \in \mathbb{F}_q$. We note these equations

$$\{C_{ik\ell h} : 0 \le i \le n-1, 0 \le k \le \psi, 0 \le \ell \le \gamma, 1 \le h \le \omega\} \tag{6}$$

   form a linearly independent set, since the rank of $\{(1, \vec{v_1}), \ldots, (1, \vec{v_\omega})\} \subseteq (\mathbb{F}_q)^{\omega+1}$ is ω.

2. The signing information for each $U_h \in C$. That is, the initial signing keys $sk_h^0(\vec{y}, z) = F(U_h, \vec{y}, z)$, and the master signing keys $mk_h(\vec{y}, z, t) = mk(U_h, \vec{y}, z, t)$.
Rewriting these equations, we have

$$sk_h^0(\vec{y}, z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} U_h^i a_{ijk0} y_j z^k,$$

and

$$mk_h(\vec{y}, z, t) = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=1}^{\gamma} U_h^i a_{ijk\ell} y_j z^k t^\ell.$$

Note that sk_h^0 and mk_h are polynomials with terms of the form $(d_{jk\ell})_h y_j z^k t^\ell$, for $0 \le j \le \omega$, $0 \le k \le \psi$, and $0 \le \ell \le \gamma$. So C has access to $(\omega+1)(\psi+1)(\gamma+1)$ equations $D_{jk\ell h}$ in the unknown coefficients $a_{ijk\ell}$ for each $U_h \in C$, where

$$D_{jk\ell h} \colon \sum_{i=0}^{n-1} U_h^i a_{ijk\ell} = (d_{jk\ell})_h$$

for some (known) $(d_{jk\ell})_h \in \mathbb{F}_q$. Now, these equations, together with the equations from (6), are not a linearly independent set, due to the relationships between users' signing and verification keys. More specifically, for any users U_h and U_h', we have

$$sk_h^0(\vec{y}, z)\big|_{\vec{y}=v_{h'}^{\rightarrow}} + mk_h(\vec{y}, z, t)\big|_{\vec{y}=v_{h'}^{\rightarrow}} = \tilde{v}_{h'}(x, z, t)\big|_{x=U_h}. \tag{7}$$

Equation (7) implies that for each $U_h \in \mathcal{C}$ and each choice of k and ℓ, for $0 \le k \le \psi$ and $0 \le \ell \le \gamma$, we have a set of ω relations among the $\omega+1$ equations $\{D_{jk\ell h} : 0 \le j \le \omega\}$. Thus, the information gleaned from the coalition's signing information is contained in the set

$$\{D_{0k\ell h} : 0 \le k \le \psi, 0 \le \ell \le \gamma, 1 \le h \le \omega\}. \tag{8}$$

3. Key exposure information for honest users. For each $U_h \notin C$, we allow the coalition either signing key exposure or master key exposure (but not both for a given user). For a given $U_h \notin C$, this information takes one of the following forms.
   - Signing key exposure for $U_h \notin C$:
   C has access to $\mathsf{Sign}_h^{t_{h_1}}, \ldots, \mathsf{Sign}_h^{t_{h_\gamma}}$, where $t_{h_1}, \ldots, t_{h_\gamma}$ are valid time periods. We have, for a given time period t_{h_d} (where $1 \le d \le \gamma$),

$$\begin{aligned}\mathsf{Sign}_h^{t_{h_d}}(\vec{y}, z) &= G(U_h, \vec{y}, z, t_{h_d}) \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=0}^{\gamma} a_{ijk\ell} U_h^i \left(t_{h_d}\right)^\ell y_j z^k.\end{aligned}$$

   Note that $\mathsf{Sign}_h^{t_{h_d}}$ is a polynomial with terms of the form $(e_{jk})^{t_{h_d}} y_j z^k$ for $0 \le j \le \omega$ and $0 \le k \le \psi$, so the coalition C has access to equations $E_{jkt_{h_d}}$ in the unknown

coefficients $a_{ijk\ell}$, where

$$E_{jkt_{h_d}} \colon \sum_{i=0}^{n-1} \sum_{\ell=0}^{\gamma} a_{ijk\ell} U_h^i \left(t_{h_d}\right)^\ell = \left(e_{jk}\right)^{t_{h_d}}$$

for some (known) $\left(e_{jk}\right)^{t_{h_d}} \in \mathbb{F}_q$. In a manner similar to the previous analysis, we observe the relation

$$\mathsf{Sign}_h^{t_{h_d}}(\vec{y}, z)\Big|_{\vec{y}=\vec{v_{h'}}} = \tilde{v}_{h'}(x, z, t)|_{x=U_h, t=t_{h_d}}$$

for any pair of users U_h and $U_{h'}$. Thus, considering $U_{h'} \in C$ and fixing k, we have a set of ω relations among the $\omega + 1$ equations $\left\{E_{jkt_{h_d}} : 0 \leq j \leq \omega\right\}$. This implies that any new information gained by signing key exposure for the user U_h is contained in the set

$$\left\{E_{0kt_{h_d}} : 0 \leq k \leq \psi, 1 \leq d \leq \gamma\right\}. \tag{9}$$

- Master key exposure for $U_h \notin C$:

$$mk_h(\vec{y}, z, t) = \sum_{i=0}^{n-1} \sum_{j=0}^{\omega} \sum_{k=0}^{\psi} \sum_{\ell=1}^{\gamma} U_h^i a_{ijk\ell} y_j z^k t^\ell.$$

Now, mk_h is a polynomial with terms of the form $\left(d_{jk\ell}\right)_\ell y_j z^k t^\ell$, for $0 \leq j \leq \omega$, $0 \leq k \leq \psi$, and $1 \leq \ell \leq \gamma$. So C has access to $(\omega + 1)(\psi + 1)(\gamma)$ equations $D_{jk\ell h}$ in the unknown coefficients $a_{ijk\ell}$ for each $U_h \notin C$, where

$$D_{jk\ell h} \colon \sum_{i=0}^{n-1} U_h^i a_{ijk} = \left(d_{jk\ell}\right)_h$$

for some (known) $\left(d_{jk\ell}\right)_h \in \mathbb{F}_q$. As before, the relation between users' signing and verification keys implies that it suffices to consider the set

$$\left\{D_{0k\ell h} : 0 \leq k \leq \psi, 1 \leq \ell \leq \gamma\right\} \tag{10}$$

for the user U_h.

4. Signing oracle queries for each user $U_h \notin C$. The coalition C has access to ψ signing oracle queries for each user $U_h \notin C$. We first observe that for both types of key exposure, the result of a signing oracle query on message m for a period t_0 contains enough information for C to determine the signature on m for all other time periods. This is easy to see for the case of master key exposure, since C can compute $F(x, y, z)|_{x=U_h, z=m}$ by subtracting $mk_h(\vec{y}, z, t)|_{z=m, t=t_0}$ from the signature. This is the case for signing key exposure so long as the coalition C maximizes its information by requesting a signature for a time period for which C does not already have the signing key. In this case, C knows a signature on m in $\gamma + 1$ time periods, so he can solve for $G(x, \vec{y}, z, t)|_{x=U_h, z=m}$, as this is a polynomial of degree γ in t. Therefore the time period requested in a signature oracle query is irrelevant to this analysis; for simplicity

we use t_h as a placeholder for the time period in requested signatures from signer U_h.

That is, C has access to up to ψ signatures $\sigma_{h,k'}$ from each user $U_h \notin C$, on messages $m_{h,k'}$ of C's choice, where $1 \le k' \le \psi$, with the exception that C can only access a signature $\sigma_{\zeta,k'}$ on a message $m_{\zeta,k'} \neq m$ with signer U_ζ.
Each requested signature has the form

$$\begin{aligned}\sigma_{h,k'} &= G(x,\vec{y},z,t)\big|_{x=U_h,z=m_{h,k'},t=t_h} \\ &= \sum_{i=0}^{n-1}\sum_{j=0}^{\omega}\sum_{k=0}^{\psi}\sum_{\ell=0}^{\gamma} a_{ijk\ell}U_h^i\left(m_{h,k'}\right)^k\left(t_h\right)^\ell y_j.\end{aligned}$$

Note that $\sigma_{h,k'}$ is a polynomial with terms of the form $\left(b_j\right)_{h,k'} y_j$ for $0 \le j \le \omega$, so C has access to equations $B_{jhk'}$, where

$$B_{jhk'}\colon \sum_{i=0}^{n-1}\sum_{k=0}^{\psi}\sum_{\ell=0}^{\gamma} a_{ijk\ell}U_h^i\left(m_{h,k'}\right)^k\left(t_h\right)^\ell = \left(b_j\right)_{h,k'}$$

for some (known) $\left(b_j\right)_{h,k'} \in \mathbb{F}_q$. As before, we have

$$\sigma_{h,k'}^{t_h}(\vec{y})\big|_{\vec{y}=v_{h'}^{\vec{}}} = \tilde{v}_{h'}(x,z,t)\big|_{x=U_h,z=m_{h,k'},t=t_h}$$

for each $U_{h'} \in C$. Thus it suffices to consider the set

$$\{B_{0hk'} : 1 \le k' \le \psi\} \tag{11}$$

for each $U_h \notin C$.

5. Up to ψ_V query results from the oracle $\mathsf{Vrfy}_h^{\mathcal{O}}$ for $U_h \notin C$. In the following, we discuss the attack scenario without $\mathsf{Vrfy}^{\mathcal{O}}$ queries. Incorporating these queries into the analysis follows as in the proof of Theorem 2.11.

To summarize, the information obtained by the coalition C is contained in the following sets of equations: sets (6) and (8), together with, for each $U_h \notin C$, one of set (9) or set (10) (depending on the type of key exposure), and set (11). These equations do form a linearly independent set; we provide the proof in Appendix A.2. We have a total of $n\omega(\psi+1)(\gamma+1)+\omega(\psi+1)(\gamma+1)+(n-\omega)\gamma(\psi+1)+(n-\omega)\psi$ equations, which implies that we have $n-\omega$ free variables in the given linear system.

With the given information, C can consider the polynomials $G'(x,\vec{y},z,t)$ consistent with the known information about $G(x,\vec{y},z,t)$. If a given polynomial G' is consistent with the known information about G, we say G' satisfies property $(*)$. We let

$$\mathcal{G} = \{G'(x,\vec{y},z,t) : G' \text{ satisfies } (*)\}.$$

From above, we have $|\mathcal{G}| = q^{n-\omega}$.

Case 1: $U_\zeta \notin C$, $U_\nu \in C$

In this case, the goal of C is to produce a ζ-authentic signature (m, σ) for some time period t' (for which C does not already have the corresponding signing key). We first observe that producing a ζ-authentic signature for such a time period t' is equivalent to producing U_ζ's general signing key $\mathsf{Sign}_\zeta(\vec{y}, z, t) = G(x, \vec{y}, z, t)|_{x=U_\zeta}$. Once we have this result, the rest of the proof for this case is almost identical to that provided in Theorem 2.11, so we do not provide the details here.

To see this, suppose C produces a ζ-authentic signature (m, σ) for time period $t' \neq t_{\zeta_i}$ for $1 \leq i \leq \gamma$ consistent with C's information. Then C has access to a total of $\psi + 1$ points (namely, the signatures on m and on $m_{\zeta,1}, \ldots, m_{\zeta,\psi}$) on $G(x, \vec{y}, z, t)|_{x=U_\zeta, t=t'}$, which is a polynomial of degree ψ in z. Thus C can solve for

$$\mathsf{Sign}_\zeta^{t'}(\vec{y}, z) = G(x, \vec{y}, z, t)|_{x=U_\zeta, t=t'},$$

so C knows U_ζ's signing key for the time period t'. There are then two cases to consider, depending on which type of key exposure C has for the target signer U_ζ:

1. Suppose C has signing key exposure against U_ζ for γ time periods $t_{\zeta_1}, \ldots, t_{\zeta_\gamma}$. Then C knows $\mathsf{Sign}_\zeta^{t'}, \mathsf{Sign}_\zeta^{t_{\zeta_1}}, \ldots, \mathsf{Sign}_\zeta^{t_{\zeta_\gamma}}$, i.e., $\gamma+1$ points on $\mathsf{Sign}_\zeta(\vec{y}, z, t)$, which is a polynomial of degree γ in t. So C can solve for $\mathsf{Sign}_\zeta(\vec{y}, z, t)$.
2. Suppose C has master key exposure for U_ζ, so C knows $mk_\zeta(\vec{y}, z, t)$. Then

   $$\mathsf{Sign}_\zeta^{t'}(\vec{y}, z) - mk_\zeta(\vec{y}, z, t)|_{t=t'} = (G - mk)(x, \vec{y}, z, t)|_{x=U_\zeta, t=t'} = F(x, \vec{y}, z)|_{x=U_\zeta}.$$

   That is, C knows both $F(x, \vec{y}, z)|_{x=U_\zeta}$ and $mk_\zeta(\vec{y}, z, t)$, the sum of which yields $\mathsf{Sign}_\zeta(\vec{y}, z, t)$.

Case 2: $U_\zeta \notin C$, $U_\nu \notin C$ and *Case 3: $U_\zeta \in C$, $U_\nu \notin C$*

The case where $U_\zeta \notin C$, $U_\nu \notin C$ and the case where $U_\zeta \in C$, $U_\nu \notin C$ follow the same argument as for the basic USS scheme provided in the proof of Theorem 2.11, so we do not reproduce the proof here. □

Theorem 2.13. *The above scheme has secure key updates.*

Proof. This is easy to see from the scheme definition. For a given user U_h, consider the signing information $\mathsf{Sign}_h^{t_{h_1}}$ and $\mathsf{Sign}_h^{t_{h_2}}$ from the two consecutive periods t_{h_1} and t_{h_2}. We see that

$$\mathsf{Sign}_h^{t_{h_2}} - \mathsf{Sign}_h^{t_{h_1}} = mk_h^{(t_{h_1}, t_{h_2})},$$

which is the key-updating information from period t_{h_1} to t_{h_2}. □

2.10. Discussion and Comparison with Related Work

We have discussed related work in some detail throughout the chapter, but in this section, we give a brief overview of the field and touch on the differences between our work and that in the literature. As mentioned in Section 2.1, there has been a lot of research devoted to constructing unconditionally secure signatures since Chaum and Roijakkers [8] first introduced the concept. A popular approach has been to enhance existing unconditionally secure message authentication codes (MACs) [38, 39, 44, 58] in order to ensure non-repudiation, transferability, and unforgeability are satisfied.

Recently, Roeder et al. [57] introduced the notion of *multi-verifier signatures*, which are somewhat similar to unconditionally secure signatures in flavor; however, they are only computationally secure. Roeder et al. explore using enhanced MACs to create more efficient signature schemes in the computational setting, as a result proposing multi-verifier signatures. The basic idea is that a signer has pairwise shared keys with a finite number of verifiers, but does not know which key is shared with which verifier. This prevents the signer from knowing the identity of the verifier who would accept a particular fraudulent signature.

In the unconditionally secure setting, much of the work on security models for USS schemes [38, 39, 44, 58] draws upon standard MAC security models, with only one [65] drawing on notions from traditional public-key cryptography. In comparison to other works, our approach is most similar to that of Shikata et al. [65], whose model is also designed as an extension of public-key signature security notions. We compare our model with that of Shikata et al. in Section 2.10.1. Our model differs from those in the existing literature in its careful treatment of ζ-authentic and (ζ, ν)-fraudulent signatures. Moreover, our treatment of dispute resolution for USS schemes goes far beyond previous work, as no other paper analyzes dispute resolution methods in any detail or formally defines non-repudiation and transferability.

The Hara et al. model [40] for unconditionally secure blind signatures is essentially the same as the Shikata et al. model [65] with an added blindness condition. Hara et al. separate the unforgeability definition of Shikata et al. into a weaker notion of unforgeability and an additional non-repudiation requirement. The non-repudiation requirement actually treats more cases than a simple non-repudiation attack (as the success of the attack is not dependent on dispute resolution), so the reason for this separation is unclear. Hara et al. also allow the signer to be the target verifier, which is not explicitly allowed in the Shikata et al. model, and so they add a separate unforgeability definition for this case.

The models of Hanaoka et al. [38, 39] and Safavi-Naini et al. [58] are based on security notions from message authentication codes (MACs). Hanaoka et al. treat only a limited attack scenario (which is covered by our model), including *impersonation*, *substitution*, and *transfer with a trap*, and they do not include a verification oracle. Safavi-Naini et al. treat a similar range of attacks as our model, specified through *denial*, *spoofing*, and *framing*

attacks, and allow both signature and verification oracles, but do not make the distinction between ζ-authentic and (ζ, ν)-fraudulent signatures. Furthermore, our model is more concise, as the denial attack covers a signer trying to repudiate a signature, whereas we show that it is unnecessary to treat non-repudiation as a separate part of an unforgeability definition. In addition, not all attack scenarios included in our definition are covered by the Safavi-Naini et al. model. For instance, the attack consisting of signer $U_\zeta \in C$ with target verifier U_ν, where C creates a (ζ, ν)-fraudulent signature, is not considered. The Safavi-Naini et al. model considers this scenario only in the case where an arbiter is involved and rejects the signature (i.e. a denial attack). In certain applications (e.g., e-cash) we do not want the signer to be able to create a (ζ, ν)-fraudulent signature, regardless of whether a dispute resolution mechanism is invoked.

The concept of key insulation was introduced for traditional public-key cryptosystems and digital signatures by Dodis et al. [**23**, **24**]. The definition given for KI-USS schemes in Section 2.8 draws from the work Seito et al. [**63**] and Seito and Shikata [**62**] on unconditionally secure key-insulated multi-receiver authentication codes and key agreement, respectively. Our definition is consistent with the work of Dodis et al. [**24**] on key-insulated signatures in the traditional public-key cryptography setting. Where possible, we have simplified the definitions and notation. For example, we assume key-updating occurs once every time period and uses information on the insecure device from the *previous* time period. The definitions given by Seito et al. [**63**] and Seito and Shikata [**62**] do not specify the use of consecutive time periods for key-updating. The model for KI-USS schemes given in Section 2.8 incorporates the notion of strong key insulation introduced by Dodis et al. [**23**] and is a natural extension of our basic USS security model.

The proof of security for Hanaoka et al.'s [**38**] construction given in Section 2.7 is a useful addition to the literature, given the tendency to forego security proofs due to lack of space. The security analysis is particularly useful in understanding the motivation behind the given multivariate polynomial construction. Moreover, this construction, combined with the techniques used by Seito et al [**63**] for key-insulated multi-receiver authentication codes, lends itself quite naturally to the KI-USS scheme presented in Section 2.9. This method of constructing the KI-USS scheme, which we discuss in more detail in the relevant section, has the nice consequence that the security argument reduces to the security argument for the basic USS scheme.

2.10.1. Comparison with Shikata et al.'s model. In this section, we discuss several aspects of the model of Shikata et al. [**65**] and how our approach differs from theirs.

1. Shikata et al.'s model [**65**] is limited to a single-signer scenario. We consider a more general model in which any participant can be a signer.
2. In Definition 2 [**65**], a signed message (x, σ) is defined to be *valid* if it was created using the signer's signing algorithm. Then, in their "Requirement 1", which includes notions for verifiability, dispute resolution, and unforgeability, it is stated that (x, σ)

is valid if and only if U_ν's verification algorithm outputs *True* when given (x, σ) as input. This requirement is problematic, since U_ν can use knowledge of his verification algorithm to find a pair (x, σ) that has output *True*; such a pair is then "valid." However, this means that a receiver can create valid signatures, and consequently the signature scheme does not provide unforgeability. Shikata et al. relax this condition in Requirement 2 by allowing a small error probability that an "invalid" signature is accepted by a given verifier. However, this does not rectify the aforementioned problem, as the probability space in this definition is unspecified.

3. Shikata et al.'s [65] definitions of *existential forgery* and *existential acceptance forgery* (Definitions 3 and 4, respectively) are rather complicated. It seems that the notion of "existential forgery" corresponds to our definition of a ζ-*authentic signature*. The coalition that creates this signature should not include U_ζ. The notion of "existential acceptance forgery" apparently is dependent upon the coalition that creates it. If U_ζ is in the coalition, then an existential acceptance forgery would most naturally coincide with our definition of a (ζ, ν)-*fraudulent signature*. If U_ζ is not in the coalition, then it would more likely mean a (ζ, ν)-*acceptable signature*. In each case, the coalition creating the signature should not include U_ν. These definitions are a bit confusing, and we believe that the concepts of authentic, acceptable, and fraudulent signatures are helpful in phrasing clear and concise definitions.
4. In Theorem 2 [65], it is stated without proof that a signature scheme that is "existentially acceptance unforgeable" is necessarily "existentially unforgeable." Roughly speaking, this is logically equivalent to the statement that an adversary that can create an existential forgery can also create an existential acceptance forgery. This statement seems rather obvious, but we need to also consider the coalitions that are creating these signatures. The adversary creating the existential forgery (i.e., a ζ-authentic signature) could be any coalition C that does not include U_ζ. A ζ-authentic signature is an existential acceptance forgery for any user $U_\nu \notin C \cup \{U_\zeta\}$. However, a problem arises if C consists of all users except for U_ζ. In this situation, a ζ-authentic signature created by C is not an existential acceptance forgery for any user. This situation is not accounted for in Theorem 2 of Shikata et al.'s work [65], and therefore it does not suffice to consider only existential acceptance forgeries. We remark that our approach is consistent with that used to define A^2-codes [67], in which neither the sender nor the receiver is trusted, and so attacks solely against a target signer are considered. To be specific, Simmons [67] treats R_0 attacks, impersonation by the receiver, and R_1 attacks, substitution by the receiver. Allowing attacks in which all verifiers collude against a target signer is a generalization of this approach.
5. Notwithstanding the previous points, the definition of "strong security" by Shikata et al. [65] (Definition 9) is very similar to our properties 4(a) and 4(b) of Definition 2.5, except that Definition 9 only covers existential acceptance forgeries. In order to compare our model with that of Shikata et al. [65], we consider the following three attack scenarios, where U_ζ denotes the signer and U_ν denotes a verifier:

case A: Neither U_ζ nor U_ν is in the coalition C, and C creates a (ζ, ν)-fraudulent signature.
case B: U_ζ is not in the coalition C, and C creates a ζ-authentic signature.
case C: $U_\zeta \in C$, $U_\nu \notin C$, and C creates a (ζ, ν)-fraudulent signature.
In our security definition (Definition 2.5), property 4(a) is equivalent to the union of case A and case C, and property 4(b) is equivalent to case B. Now, Definition 9 [65] considers two attacks: property 1) is the union of cases A and B, but does not include the case where there is no target verifier, as discussed in the previous point; and property 2) is case C.

6. Finally, we give a more complete treatment of dispute resolution than is presented by Shikata et al. [65].

2.11. Concluding Remarks and Future Work

We have presented a new security model for unconditionally secure signature schemes, one which fully treats the implications of having multiple verification algorithms. In particular, we have given a formal discussion of dispute resolution, a necessary component of any USS scheme, and analyzed the effect of dispute resolution on unforgeability. We have provided formal definitions of non-repudiation and transferability, and given sufficient conditions for a USS scheme to satisfy these properties. Moreover, we have analyzed the trust assumptions required in typical examples of dispute resolution. We have given an analysis of Hanaoka et al.'s construction [38] in our security model. Finally, we have provided an extension of our basic framework to the setting of key insulation and presented a construction, inspired by the original construction of Hanaoka et al. [38] and the work of Seito et al. [63] and Seito and Shikata [62], which satisfies a restricted version of our security definitions.

One possible avenue for future work is to construct a KI-USS scheme that meets our full security definition. In addition, it may be an interesting problem to extend the notion of key insulation to incorporate verification keys as well as signing keys. In the current work, we protect against exposure of each user's signing keys, but we do not protect against exposure of each user's verification keys. Given the nature of unconditionally secure signatures, protecting against the loss of verification keys is advisable.

Another potential extension is in the area of unconditionally secure blind signatures, in which the signer does not know what message he is signing. Practical applications for blind signatures include e-cash, in which a user would like the bank to validate his e-coin (i.e., sign the e-coin), but does not want the bank to be able to trace his spending. Current work in this area [40] is built on earlier security models for USS schemes and hence does not treat the issues described above. The existing scheme, moreover, is complicated and difficult to analyze. Defining an appropriate dispute resolution mechanism in this setting, moreover, is an interesting intellectual exercise, as it is likely that multiple calls to dispute resolution should be allowed without initiating a scheme reset.

CHAPTER 3

Extended Combinatorial Constructions for Peer-to-peer User-Private Information Retrieval

3.1. Introduction

We consider the case of a user who wishes to maintain privacy when requesting documents from a database. One existing method to address this problem is *private information retrieval (PIR)* [9]. In PIR, the content of a given query is hidden from the database, but the identity of the user making the query is not protected. In our work, we focus on an interesting alternative to PIR dubbed *user-private information retrieval (UPIR)*, introduced by Domingo-Ferrer et al. [28]. UPIR, however, is only nominally related to PIR, in that it seeks to provide privacy for users of a database. In UPIR, the database knows which records have been retrieved, but does not know the identity of the person making the query. The problem that we address, then, is how to disguise user profiles from the point of view of the database. Moreover, UPIR is a method to solve this privacy problem in a manner independent of the database. We do not assume that the database is part of the scheme set-up or altering its normal behavior with respect to serving user queries.

We draw some of our terminology from Pfitzmann [53]. Here we understand *anonymity* as the state of not being identifiable within a set of subjects, and the *anonymity set* is the set of all possible subjects. By *untraceable* queries from the point of view of the database, we mean that the database cannot determine that a given set of queries belongs to the same user. One interesting caveat, which is addressed below, is that a set of queries might be deemed to come from the same user based on the subject matter of those queries. If the subject matter of a given set of queries is esoteric or otherwise unique, the database (or some other adversary) can surmise that the identity of the source is the same for all (or most) queries in this set; we call such a set of queries *linked*. In the case of linked queries, we wish to provide as much privacy as possible, in the sense that we wish the database to have no probabilistic advantage in guessing the identity of the source of a given set of linked queries. In this way, we can say the user making the linked queries still has *pseudonymity*—his identity is not known.

Much of the material in this chapter appears in the paper "Extended combinatorial constructions for peer-to-peer user-private information retrieval" [77], published in Advances in Mathematics of Communications, vol. 6, pp. 479–497 (2012).

With this terminology in mind, we might better explain UPIR as a method of database querying that is privacy-preserving and satisfies the following properties from the point of view of the database:

1. For any given user U_i, some (large) subset of all users $\mathcal{U}$ is the query anonymity set for U_i;
2. User queries are anonymous;
3. User queries are untraceable;
4. Given a set of queries that is unavoidably traceable due to subject matter, the person making the query is protected by pseudonymity.

In addition to these basic properties of user privacy against a database, we may wish to provide user privacy against other users. Ideally, a UPIR scheme provides the same privacy guarantees against other users as against the database, but as we discuss in this chapter, this usually cannot be attained in practice.

Previous work [27, 28, 71, 73] has focused on the use of a P2P network consisting of various encrypted "memory spaces" (i.e., drop boxes), to which users can post their own queries, submit queries to the database and post the respective answers, and read answers to previously posted queries. That is, in the P2P UPIR setting, we have a cooperating community of users who act as proxies to submit each other's queries to the database; the database itself is an independent entity not assumed to actively cooperate with the UPIR scheme. In particular, a class of combinatorial designs known as *configurations* (defined in Definition 1.23) have been suggested by Domingo-Ferrer, Bras-Amorós et al. [27,28,71,73] as a way to specify the structure of the P2P network. In this work, we focus on P2P UPIR and consider the application of other types of designs in determining the structure of the P2P network. We introduce new P2P UPIR protocols and explore the level of privacy guarantees our protocols achieve, both against the database and against other users.

3.2. Overview of Contributions

The main contributions of our work are as follows.

- We establish a strengthened model for P2P UPIR and clarify the privacy goals of such schemes using standard terminology from the field of privacy research.
- We provide an analysis of the protocol introduced by Domingo-Ferrer and Bras-Amorós [27, 28], as well as its subsequent variations. In particular, we reconsider the choice to limit the designs used as the basis for the P2P UPIR scheme to configurations. We provide a new attack on user privacy against the database, which we call the *intersection attack*, to which the above protocol variations are vulnerable.
- We introduce two new P2P UPIR protocols (and variations on these), and give an analysis of the user privacy these protocols provide, both against the database and

against other users. Our protocols utilize more general designs and resist the intersection attack by the database. In particular, our protocols provide more flexibility in designing the P2P network.

- We consider the possible trade-offs of using different types of designs in the P2P UPIR setting, both with respect to the overall flexibility of the scheme as well as user privacy. Our protocols provide viable design choices, which can allow for a *dynamic UPIR scheme* (i.e., one in which users are permitted to enter and leave the system), or provide increased privacy against other users.
- We consider the problem of user privacy against other users in detail. In particular, we relax the assumptions of previous work, by allowing users to collaborate outside the parameters of the P2P UPIR scheme; that is, we consider a stronger adversarial model than previous work. We analyze the ability of different types of designs to provide user privacy against other users, and explore how well our protocol resists an intersection attack launched by a coalition of users on linked queries. Finally, we introduce methods to improve privacy against other users without compromising privacy against the database.

3.2.1. Chapter outline. For terminology related to combinatorial designs, we refer the reader to Section 1.2. In Section 3.3, we give a model for P2P UPIR schemes and provide the relevant privacy goals. We then review previous work in Section 3.4 and give attacks on these protocols in Section 3.4.1. We introduce our protocols in Section 3.5 and give an analysis of the privacy guarantees our protocols provide against the database. In Section 3.6, we analyze the ability of our protocols to provide user privacy against other users and consider ways to improve this type of privacy. We discuss related work in Section 3.7 and we conclude in Section 3.8.

3.3. Our P2P UPIR Model

A *P2P UPIR scheme* consists of the following players: a finite set of possible *users* $\mathcal{U} = \{U_1, \ldots, U_v\}$, the *target database* DB, and an *external observer*, O. We assume all communication in a P2P UPIR scheme is encrypted, including communication between the users and DB.

In a basic P2P UPIR scheme, users have access to secure drop boxes known as *memory spaces*. More precisely, a *memory space* is an abstract (encrypted) storage space in which some subset of users can store and extract queries and query responses; the exact structure of these spaces is not specified. We let $\mathcal{S} = \{S_1, \ldots, S_b\}$ denote the set of memory spaces, and we let K_i denote the (symmetric) key associated with S_i, for $1 \leq i \leq b$. We assume that encryption keys for memory spaces are only known to a given subset of users, as specified by the P2P UPIR protocol. For the sake of simplicity, we assume that these keys are initially distributed in a secure manner by some trusted external entity (not the database DB). However, the precise method by which these keys are distributed is not relevant to

the results we prove in this research. Similarly, the choice of symmetric encryption scheme used in the P2P UPIR scheme is not relevant; we assume computational security for this aspect of the scheme. If two distinct users U_i, $U_j \in \mathcal{U}$ have access to a common memory space, then we say U_i and U_j are *neighbors*. Similarly, the *neighborhood* of a user U_i is defined as the set of all neighbors of U_i, and it is denoted as $N(U_i)$.

When a user U_i wishes to send a query q to DB, we say U_i is the *source* of the query. Rather than sending the query directly to DB, the user U_i writes an encrypted copy of q, together with a requested proxy U_j, to a memory space S_ℓ. Here U_j is the *proxy* for U_i's query q, and consequently U_j must know the encryption key K_ℓ corresponding to the memory space S_ℓ. The user U_j decrypts the query, re-encrypts q under a secret key shared with DB, say K^j_{DB}, and forwards this re-encrypted query $e_{K^j_{DB}}(q)$ to DB. DB sends back a response, which U_j first decrypts, then re-encrypts under K_ℓ and records in the memory space S_ℓ. We give a schematic of the information flow of a basic P2P UPIR scheme in Figure 1.

FIGURE 1. Schematic of Information Flow

User U_i		**Memory Space** S_ℓ		**Proxy** U_j		**DB**
query q and proxy U_j	$\xrightarrow{e_{K_\ell}(q), U_j}$	$(e_{K_\ell}(q), U_j)$ stored	$\xrightarrow{e_{K_\ell}(q)}$	reads q from S_ℓ	$\xrightarrow{e_{K^j_{DB}}(q)}$	processes q
reads r from S_ℓ	$\xleftarrow{e_{K_\ell}(r)}$	$e_{K_\ell}(r)$ stored	$\xleftarrow{e_{K_\ell}(r)}$	receives r from DB	$\xleftarrow{e_{K^j_{DB}}(r)}$	response r

3.3.1. Attack model. We consider each type of player as a possible adversary $\mathcal{A}$. We assume that $\mathcal{A}$ has full knowledge of the P2P UPIR scheme specification, including any public parameters, as well as any secret information assigned to $\mathcal{A}$ as part of the P2P UPIR scheme. In addition, we assume $\mathcal{A}$ does not conduct traffic analysis. The following definition is useful.

Definition 3.1. Consider a set of one or more users C. *The query sphere for* C is the set of memory spaces that C can (collectively) access via the P2P UPIR scheme.

In addition to the above, we make the following assumptions about each specific type of adversary $\mathcal{A}$:

- Suppose $\mathcal{A}$ is the database DB. As stated above, we assume that DB does not observe information being posted to or read from memory spaces. In addition, we assume that DB does not collaborate with any users and answers queries honestly. We note DB necessarily observes the content of all queries and the proxy of each query.

- Suppose $\mathcal{A}$ consists of a user or a subset of colluding users $C \subset \mathcal{U}$. We assume users are honest-but-curious. Users in C can communicate outside of the given P2P UPIR scheme and collaborate using joint information. The users of C can see the content of any queries within C's query sphere, but cannot identify the original source of these queries.
- Suppose $\mathcal{A}$ is an external adversary O. An external observer O can see the encrypted content of memory spaces. We consider the possibility of *key leakage* as the main attack launched by O. This refers to a party gaining access to a memory space key outside of the P2P UPIR scheme specification (e.g., by social engineering or other means).

Although we do not specifically treat traffic analysis as an attack, we wish to avoid a trivial analysis of traffic entering and leaving a given memory space. That is, we assume that memory spaces have encryption and decryption capabilities, so that a user acting as a proxy may decrypt and re-encrypt a given query within its associated memory space, before forwarding the query to the database.

3.3.2. Privacy and adversarial goals in P2P UPIR. In considering the privacy guarantees for a user U_i, we assume either the database DB or a group of other users may try to determine whether U_i is the source of a given set of queries, or try to establish whether or not a given set of queries originates from the same source. We need the following definition:

Definition 3.2. We say two or more queries $q_1, q_2, \ldots$ are *linked* if, given the subject matter, one can infer that the queries are likely to be from the same source.

We also consider the possibility of an external observer O gaining information that compromises the privacy of U_i. We recognize the following goals for U_i's privacy:

- *Confidentiality*: the content of U_i's queries is protected;
- *Anonymity*: the identity of a query source is protected;
- *Untraceability*: a user's query history cannot be reconstructed as having originated from the same user;
- *Pseudonymity in the presence of linked queries*: given a set of linked queries, the identity of the source is protected.

We now analyze each type of adversary $\mathcal{A}$ with respect to the above privacy goals:

- Suppose $\mathcal{A}$ is the database DB. We are not concerned with confidentiality against DB, but rather anonymity, untraceability, and pseudonymity in the presence of linked queries. The goal of the database is to create a profile of U_i. That is, the database wants to establish the set of queries for which U_i is the source. The database also attempts to trace user query histories; that is, DB wants to establish that a given set of queries came from the same source, even if DB cannot determine the identity of the source.

- Suppose $\mathcal{A}$ consists of a user or a subset of colluding users $C \subset \mathcal{U}$. The coalition C collaborates to try to determine the query history of another user $U_i \notin C$. Here we are interested in maintaining anonymity, untraceability, and pseudonymity in the presence of linked queries against C. We are also interested in maintaining confidentiality, in the sense that C should not have access to the content of queries outside the query sphere for C.
- Suppose $\mathcal{A}$ is an external adversary O. The goal of O is to compromise both the confidentiality and the anonymity of U_i. External adversaries may try to compromise the encryption mechanism of the memory spaces.

In the rest of this chapter, we focus on the database DB and user coalitions as adversaries. In particular, we are concerned with ensuring unconditional security with respect to the properties of anonymity, untraceability, and pseudonymity.

We are now almost ready to consider the P2P UPIR protocols of Domingo-Ferrer, Bras-Amorós et al. [27, 28], as well as the subsequent modification of Stokes and Bras-Amorós [71, 73]. Both these protocols and ours draw heavily from the field of combinatorial designs. The requisite background knowledge on combinatorial designs is presented in Section 1.2.

3.3.3. P2P UPIR using combinatorial designs. We model a P2P UPIR scheme using a combinatorial design. That is, we consider pairs $(\mathcal{U}, \mathcal{S})$ (where as before, $|\mathcal{U}| = v$ and $|\mathcal{S}| = b$), such that each memory space, or *block*, consists of k users and each user, or *point*, is associated with r memory spaces. That is, we assume that the pair $(\mathcal{U}, \mathcal{S})$ is a (v, b, r, k)-1-design.

We can also view the b memory spaces as points and define v blocks, each of which contains the memory spaces to which a given user belongs. This yields the dual design $(\mathcal{S}, \mathcal{U})$, which is a (b, v, k, r)-1-design.

Domingo-Ferrer, Bras-Amorós et al. [27, 28] consider trivial solutions to P2P UPIR, namely those satisfying $b = 1$ or $b = \binom{v}{2}$, in great detail. In the first case, we have a single shared memory space to which all users have access. While this solution has certain advantages with respect to privacy in front of other users (in that users have the same view of the network as the database), the disadvantages include a loss of confidentiality in front of other users, lack of scalability, and increased likelihood of key leakage. In the second trivial solution, each pair of users shares a unique memory space. This solution makes key leakage and loss of confidentiality in front of other users less of a concern, but is less efficient in terms of key management and network performance, and has the disadvantage that the two users belonging to a shared memory space necessarily know the identity of the source whenever this memory space is used. In particular, these reasons motivate the use of more general designs to specify the network structure.

3.4. Previous Work: Using Configurations

We briefly review the P2P UPIR scheme proposed by Domingo-Ferrer et al. [28] and the proposed modification of Stokes and Bras-Amorós [73]. We fix a (v, b, r, k)-configuration $(\mathcal{U}, \mathcal{S})$. As before, we have a finite set of users $\mathcal{U} = \{U_1, \ldots, U_v\}$, a database DB, and a finite set of memory spaces $\mathcal{S} = \{S_1, \ldots, S_b\}$.

Each user has access to r memory spaces, and each memory space is accessible to k users. Each memory space is encrypted via a symmetric encryption scheme; for each memory space, only the k users assigned to that memory space are given the key. The following protocol [28] assumes the user U_i has a query to submit to the database:

Protocol 1. *Domingo-Ferrer–Bras-Amorós–Wu–Manjón (DBWM) Protocol*
We fix a (v, b, r, k)-configuration.

1. The user U_i randomly selects a memory space S_ℓ to which he has access
2. The user U_i decrypts the content on the memory space S_ℓ using the corresponding key. His behavior is then determined by the content on the memory space as follows:
   (a) The content is garbage. Then U_i encrypts his query and records it in S_ℓ.
   (b) The content is a query posted by another user. Then U_i forwards the query to the database and awaits the answer. When U_i receives the answer, he encrypts it and records it in S_ℓ. He then restarts the protocol with the intention to post his query.
   (c) The content is a query posted by the user himself. Then U_i does not forward the query to the database. Instead U_i restarts the protocol with the intention to post his query.
   (d) The content is an answer to a query posted by another user. Then U_i restarts the protocol with the intention to post his query;
   (e) The content is an answer to a query posted by the user himself. Then U_i reads the query answer and erases it from the memory space. Subsequently U_i encrypts his new query and records it in S_ℓ.

The modification proposed by Stokes and Bras-Amorós [73] replaces 2(c) as follows:

Protocol 2. *DBWM–Stokes (DBWMS) Protocol*

2(c) If the content is a query posted by the user himself, then U_i forwards the query to the database with a specified probability p. If U_i forwards the query to the database, he records the answer in S_ℓ. The user U_i restarts the protocol with the intention to post his current query.

Remark 3.1. This protocol is ambiguous as stated by Stokes and Bras-Amorós. The intent of the system is that users periodically run the protocol with "garbage" queries, in this way collecting the answers to their previous queries. We refer the reader to the original protocol specifications [28, 73] for more details.

Stokes and Bras-Amorós [71,73] argue that the finite projective planes are the optimal configurations to use for P2P UPIR. Their argument is that privacy against the database is an increasing function of $r(k-1)$, since there are $r(k-1)$ users in the anonymity set of any given user U_i. That is, the query profile of U_i is diffused among $r(k-1)$ other users in the neighborhood of U_i. Now, since $r(k-1) \leq v-1$ in a configuration, the authors consider configurations satisfying $r(k-1) = v-1$, which yield the finite projective planes. In our protocols, introduced in Section 3.5, we also have neighborhoods of maximum size, without limiting ourselves to configurations. We also ensure that the database DB has no advantage in guessing the identity of the source of any given query.

3.4.1. Attacks. We consider the privacy properties of the DBWM and DBWMS protocols with respect to the database, before offering an improved protocol in Section 3.5. We fix a (v, b, r, k)-configuration, where v is the number of users and b is the number of memory spaces. We associate a block with each memory space, where the block consists of the users that have access to the memory space.

The weakness of the DBWM and DBWMS protocols lies in the possibility of a user's query history being identifiable as originating from one user. That is, if a series of queries is on some esoteric subject, the adversary (such as the database) can surmise that the source of these queries is the same. As before, we refer to such queries as *linked*.

Stokes and Bras-Amorós [73] noticed a weakness in the DBWM protocol when a projective plane is used as the configuration, that is, when $v = r(k-1)+1$. In this case, each user U_i has a neighborhood consisting of all other users. Then, given a large enough set of linked queries, the only user who never submits one of these linked queries is the source, U_i. Therefore, the database can eventually identify U_i as the source. Stokes and Bras-Amorós [73] introduced Protocol 2 to circumvent this attack. Subsequent to our research, Stokes and Bras-Amorós [72] noted this weakness applies more generally to $(v, k, 1)$-BIBDs.

We introduce another type of attack, which we call the *intersection attack*, in keeping with standard terminology from the field of privacy research [54]. This attack only applies to configurations satisfying $v > r(k-1)+1$, as it requires that all users have neighborhoods of cardinality less than $v-1$. The idea behind the intersection attack is that, given a query q_1 submitted by proxy U_j, an attacker can, by analyzing the neighborhood of U_j, compute a list of possible sources Q_1. If the attacker has access to a set of linked queries $q_1, q_2, \ldots, q_n$, and the neighborhoods of these users do not consist of all users in the system, the intersection of the possible source sets $Q_1, Q_2, \ldots, Q_n$ can perhaps identify the source (or narrow down the list of possible sources). We demonstrate this attack in the following example.

EXAMPLE 3.1. Suppose $v = 12$ and $b = 8$ and we have the following blocks (memory spaces):

$$\begin{array}{cccc} \{U_1, U_2, U_3\} & \{U_4, U_5, U_6\} & \{U_7, U_8, U_9\} & \{U_{10}, U_{11}, U_{12}\} \\ \{U_1, U_4, U_7\} & \{U_2, U_5, U_{10}\} & \{U_3, U_8, U_{11}\} & \{U_6, U_9, U_{12}\} \end{array}$$

Note this is a $(12, 8, 2, 3)$-configuration. We consider the DBWM protocol here; that is, we assume that the proxy of a given query is always different from the source of the query. Now suppose three queries are transmitted from users U_2, U_{11}, and U_8.

- If the proxy is U_2, then the source $U_i \in \{U_1, U_3, U_5, U_{10}\}$.
- If the proxy is U_{11}, then the source $U_i \in \{U_3, U_8, U_{10}, U_{12}\}$.
- If the proxy is U_8, then the source $U_i \in \{U_3, U_7, U_9, U_{11}\}$.

Suppose that the subject of the queries is similar, so it can be inferred that the source of the three queries is probably the same user. Then it is easy to identify the source of the queries:

$$U_i \in \{U_1, U_3, U_5, U_{10}\} \cap \{U_3, U_8, U_{10}, U_{12}\} \cap \{U_3, U_7, U_9, U_{11}\},$$

so $U_i = U_3$. Clearly, user privacy with respect to the database is not achieved here.

We do not claim that the above-described attack always works for any configuration; it is easy to come up with examples where the attack does not work. For example, suppose that $N(U_i) \cup \{U_i\} = N(U_j) \cup \{U_j\}$ for two distinct users U_i and U_j. Then it is impossible for DB to determine whether U_i or U_j is the source of a sequence of linked queries. Independently of this research, Stokes and Bras-Amorós [72] noted that by choosing the configuration carefully, it is possible to ensure the neighborhood-to-user mapping is not unique, and to guarantee a specified lower bound on the number of possible users for a given neighborhood.

Observe that the intersection attack is not useful when one uses a finite projective plane as the configuration and users are allowed to submit their own queries. This follows because, at each stage of the intersection attack, the set of possible sources includes all users in the set system. In the next section, we formalize this observation and discuss the use of more general types of designs in P2P UPIR protocols that resist the intersection attack in a very strong sense.

3.5. Using More General Designs

As observed in Section 3.4, in order to achieve user privacy with respect to the database, we need to allow users to sometimes transmit their own queries. We suggest a different solution to the problem than that given by Bras-Amorós et al., however. In particular, we see no reason to limit the P2P network topology to configurations. Bras-Amorós et al. indicate the use of configurations as a method to increase service availability and decrease the number of required keys. Indeed, configurations have been proposed as key rings in wireless sensor networks by Lee and Stinson [48] due to memory constraints of sensor nodes. However, storage constraints are not so much an issue in P2P UPIR. We therefore consider the possibility of using other types of designs.

We make use of memory spaces that "balance" proxies for every source. We suggest to use a balanced incomplete block design (BIBD) for the set of memory spaces. We show that these designs provide optimal resistance against the intersection attack.

Our scheme also differs from DBWMS in the treatment of proxies. In the previous schemes, the identity of a proxy is not specified by the source. Queries are simply forwarded to the database by the user who most recently checked the corresponding memory space. We propose that each source designates the proxy for each query. This enables us to balance the proxies for each possible source, thereby providing "perfect" anonymity with respect to the database. Moreover, we do not assume that each memory space holds only a single query; rather, we assume that memory spaces are capable of storing multiple queries.

Protocol 3. *Proxy-designated BIBD Protocol (Version 1)*
We fix a (v, b, r, k, λ)-BIBD. To submit a query, a user U_i uses the following steps:

1. With probability $1/v$, user U_i acts as his own proxy and transmits his own query to the DB.
2. Otherwise, user U_i chooses uniformly at random one of the r memory spaces with which he is associated, say S_ℓ, and then he chooses uniformly at random a user $U_j \in S_\ell \backslash \{U_i\}$. Finally, user U_i requests that user U_j act as his proxy using the memory space S_ℓ.

Protocol 4. *Proxy-designated BIBD Protocol (Version 2)*
We fix a (v, b, r, k, λ)-BIBD. To submit a query, a user U_i uses the following steps:

1. With probability $1/v$, user U_i chooses to act as his own proxy. User U_i then writes the query uniformly at random to one of the r memory spaces with which he is associated, and transmits his own query to DB.
2. Otherwise, user U_i chooses uniformly at random one of the r memory spaces with which he is associated, say S_ℓ, and then he chooses uniformly at random a user $U_j \in S_\ell \backslash \{U_i\}$. Finally, user U_i requests that user U_j act as his proxy using the memory space S_ℓ.

Remark 3.2. We note that Protocol 4 differs from Protocol 3 only in the first step.

Remark 3.3. We assume users check memory spaces regularly and act as proxies as requested within a reasonable time interval.

Remark 3.4. We make the assumption that, when a source U_i requests U_j to be his proxy, everyone in the associated memory space knows that this request has been made, but no one (except for U_i) knows the identity of the source.

Remark 3.5. The choice between Protocol 3 and Protocol 4 impacts the amount of privacy the scheme provides against other users. This is discussed in Section 3.6.

We analyze the situation from the point of view of the database. For the rest of the chapter, we let variables $\mathbf{S}, \mathbf{P}, \mathbf{M}$ be random variables denoting the source, proxy, and memory spaces, respectively.

Theorem 3.1. *From the point of view of the database, the Proxy-designated BIBD Protocols (Protocols 3 and 4) satisfy* $\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_j] = \Pr[\mathbf{S} = U_i]$ *for all* U_i, $U_j \in \mathcal{U}$.

Proof. First, the schemes ensure that $\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] = 1/v$ for all U_i, $U_j \in \mathcal{U}$. To see this, first note that U_i picks himself as the source with probability $1/v$. In Protocol 3, U_i then submits his query directly to the database. In Protocol 4, U_i picks one of the r memory spaces with which he is associated uniformly at random and then acts as his own proxy. So in both cases, we have

$$\Pr[\mathbf{P} = U_i \mid \mathbf{S} = U_i] = \frac{1}{v}.$$

Then in both protocols, with probability $(v-1)/v$, user U_i picks a memory space S_ℓ (with $U_i \in S_\ell$) uniformly at random, followed by a proxy U_j associated with S_ℓ. The probability that a fixed U_j with $i \neq j$ acts as proxy can be computed as follows.

For $i \neq j$, we have

$$\begin{aligned}\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] &= \frac{v-1}{v} \sum_{S_\ell :\, U_i, U_j \in S_\ell} \Pr[\mathbf{M} = S_\ell] \Pr[\mathbf{P} = U_j \mid \mathbf{M} = S_\ell] \\ &= \frac{v-1}{v} \sum_{S_\ell :\, U_i, U_j \in S_\ell} \frac{1}{r(k-1)} = \left(\frac{v-1}{v}\right)\left(\frac{\lambda}{r(k-1)}\right) = \frac{1}{v}.\end{aligned}$$

We see that $\Pr[\mathbf{P} = U_j] = 1/v$ for all $U_j \in \mathcal{U}$, since

$$\Pr[\mathbf{P} = U_j] = \sum_{U_i \in \mathcal{U}} \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i] = \frac{1}{v}.$$

Now we have

$$\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_j] = \frac{\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i]}{\Pr[\mathbf{P} = U_j]} = \Pr[\mathbf{S} = U_i],$$

so the identity of the proxy gives no information about the identity of the source. □

We observe that this analysis is independent of any computational assumptions, so the security is unconditional. Since we have achieved a perfect anonymity property, it follows that no information is obtained by analyzing linked queries.

EXAMPLE 3.2. To illustrate, consider a projective plane of order 2 with the following blocks:

$$\begin{array}{llll}\{U_1, U_2, U_3\} & \{U_1, U_4, U_5\} & \{U_1, U_6, U_7\} & \{U_2, U_4, U_6\} \\ \{U_2, U_5, U_7\} & \{U_3, U_4, U_7\} & \{U_3, U_5, U_6\} & \end{array}$$

We note that this is a $(7, 3, 3, 3, 1)$-BIBD. Suppose that the first query uses block $\{U_2, U_4, U_6\}$ with proxy U_4, and the second query uses block $\{U_2, U_5, U_7\}$ with proxy U_2. From the first query, DB knows that one of three blocks were used: $\{U_1, U_4, U_5\}$, $\{U_2, U_4, U_6\}$, or $\{U_3, U_4, U_7\}$. However, $\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_4] = \Pr[\mathbf{S} = U_i]$ for all possible sources U_i, so DB has no additional information about the identity of the source, given that $\mathbf{P} = U_4$. From the second query, DB knows that one of three blocks were used:

$\{U_1, U_2, U_3\}$, $\{U_2, U_4, U_6\}$, or $\{U_2, U_5, U_7\}$. Again, $\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_2] = \Pr[\mathbf{S} = U_i]$ for all possible sources U_i, so DB has no additional information about the identity of the source, given that $\mathbf{P} = U_2$. So even if DB suspects that both queries came from the same source, he has no way to identify the source.

3.5.1. Extensions. We can consider using less structured designs than BIBDs, such as pairwise balanced designs or covering designs. It turns out that we can still achieve perfect anonymity with respect to DB, because our anonymity argument remains valid provided that $\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] = 1/v$ for all U_i, $U_j \in \mathcal{U}$.

We next give a generalized protocol based on an arbitrary covering design. That is, we do not require constant block size k or constant replication number r.

Protocol 5. *Proxy-designated Covering Design Protocol (Version 1)*
We fix a covering design. To submit a query, a user U_i performs the following steps:

1. User U_i chooses the designated proxy U_j uniformly at random. If $U_i = U_j$, then U_i submits his query directly to DB and skips Step 2.
2. If $U_i \neq U_j$, then user U_i chooses uniformly at random one of the memory spaces that contains both U_i and U_j, say S_ℓ. Then U_i requests that user U_j act as his proxy using memory space S_ℓ.

Protocol 6. *Proxy-designated Covering Design Protocol (Version 2)*
We fix a covering design. To submit a query, a user U_i performs the following steps:

1. User U_i chooses the designated proxy U_j uniformly at random. (The user U_i may choose himself as the proxy U_j.)
2. User U_i chooses uniformly at random one of the memory spaces that contains both U_i and U_j, say S_ℓ. Then U_i requests that user U_j act as his proxy using memory space S_ℓ.

REMARK 3.6. If the covering design is a BIBD, then Protocol 5 is equivalent to Protocol 3 and Protocol 6 is equivalent to Protocol 4.

REMARK 3.7. We must have a covering design to ensure that a suitable memory space S_ℓ always exists in Step 2 of Protocols 5 and 6.

REMARK 3.8. As in Protocols 3 and 4, we assume users check memory spaces regularly, and act as proxies as requested within a reasonable time interval. We also assume, as before, that when source U_i requests that $U_j \neq U_i$ be his proxy, everyone in the associated memory space knows that this request has been made, but no one (except for U_i) knows the identity of the source.

Theorem 3.2. *From the point of view of the database, for a given query, the Proxy-designated Covering Design Protocols (Protocols 5 and 6) satisfy* $\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_j] = \Pr[\mathbf{S} = U_i]$ *for all* U_i, $U_j \in \mathcal{U}$.

Proof. Step 1 of both Protocol 5 and Protocol 6 ensures that $\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] = 1/v$ for all $U_i, U_j \in \mathcal{U}$. Similarly, we can see that

$$\Pr[\mathbf{P} = U_j] = \sum_{U_i \in \mathcal{U}} \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i] = \frac{1}{v}$$

for all $U_j \in \mathcal{U}$. We once again have

$$\Pr[\mathbf{S} = U_i \mid \mathbf{P} = U_j] = \frac{\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i]}{\Pr[\mathbf{P} = U_j]} = \Pr[\mathbf{S} = U_i],$$

so the identity of the proxy gives no information about the identity of the source. □

As before, we observe that this analysis is independent of any computational assumptions, so the security is unconditional. Since we have achieved a perfect anonymity property, no information is obtained by analyzing linked queries.

3.5.2. Dynamic P2P UPIR schemes. One benefit of using less structured designs than BIBDs is that the scheme can be *dynamic*. That is, we can add and remove users, which allows greater flexibility in practice.

To delete a user U_i from Protocols 5 and 6, we simply remove U_i from all the memory spaces with which he is associated. To avoid U_i from reading any more queries written to these memory spaces, we also need a rekeying mechanism to update the associated keys. The same external entity that distributed the initial set of keys could be responsible for rekeying. The end result is a covering design with one fewer users than before.

To add a user U_{new} in Protocols 5 and 6, we may use the following method. We first find $\mathcal{M} = \{S_{h_1}, \ldots, S_{h_\ell}\} \subseteq \mathcal{S}$ such that $S_{h_1} \cup \cdots \cup S_{h_\ell} = \mathcal{U}$. That is, we need a set of memory spaces whose union contains all current users. A greedy algorithm could be used to accomplish this task, although the resultant set $\mathcal{M}$ would likely not be optimal (in the sense that ℓ would likely not be as small as possible). Indeed, finding the minimum such set is NP-hard. (This is the minimum cover problem, which is problem SP5 in Garey and Johnson [**36**].)

Once we have identified a suitable set $\mathcal{M}$, we simply add U_{new} to each memory space in $\mathcal{M}$, and give U_{new} the associated keys. In addition, we need a mechanism by which to inform all users of U_{new}'s presence in the scheme. The resulting set system is still a covering design—one which contains one more user than before.

3.6. Privacy Against Other Users

In this section, we consider our Protocols 3, 4, 5, and 6 in the context of analyzing user privacy against other users. We remind the reader of Remarks 3.4 and 3.8: we assume that when a source U_i requests that U_j be his proxy, everyone in the associated memory space

knows that this request has been made, but no one (except for U_i) knows the identity of the source.

We observe that if we have the cooperation of the database DB, we can achieve privacy against other users in a computational sense by making use of standard encryption protocols. That is, if the database is willing and has established a public key, a user wishing to forward a query q to DB can first encrypt q, together with a symmetric key K_q of his choosing, under DB's public key, and then proceed with the protocol as usual. The designated proxy then forwards the entire encrypted message to DB. Upon receiving this message, DB can decrypt, compute the response r to q, and send the encryption of r under the symmetric key K_q back to the proxy, who then posts the encrypted response to the appropriate memory space as before. The users of the chosen memory space, in particular, will be unable to read the content of any queries for which they are not the source (subject to the computational security of the chosen symmetric encryption scheme used with DB). This technique then provides confidentiality against other users, so the loss of privacy against other users in UPIR is effectively circumvented.

We now analyze the privacy of a given user relative to other users of the scheme in the original UPIR specification, in which we do not assume the database DB cooperates as described above. As we see in this analysis, if we wish to provide privacy against other users, a design that has more structure than a general covering design becomes useful. In particular, we observe that the use of a regular PBD (see Definition 1.16) in Protocols 5 and 6 is desirable. It is in general difficult to provide privacy against other users, however, since by design users must be able to see the content of their associated memory spaces. In this section, we simplify the analysis by assuming that sources are equiprobable; we leave the generalization of the analysis in the absence of this assumption as future work.

It is helpful to begin with an example:

Example 3.3. Consider the projective plane from Example 3.2 and suppose we use Protocol 3. Suppose that user U_4 is requested to act as proxy for a query in memory space $\{U_1, U_4, U_5\}$ by source U_1. User U_4 knows that the source must be U_1 or U_5 (since he did not make the request himself). User U_5, however, knows that the source must be U_1 because

1. U_5 did not make the request himself, and
2. U_4 would not post a request to himself to transmit a query—he would just go ahead and transmit it himself.

We can generalize the concept from Example 3.3. Observe that in Protocols 3 and 5, the requested proxy can rule out one possible source, and anyone else in the memory space (who is not the source) can rule out two possible sources. If we consider Protocols 4 and 6, then users can rule out only one possible source (namely, themselves). That is, Protocols 4 and 6 improve the information-theoretic privacy guarantees of the scheme with respect to the

viewpoint of other users. However, we remark that in these versions, when a source acts as his own proxy, other users associated with the chosen memory space can see the content of the query. In Protocols 3 and 5, if a user U_i is both the source and proxy of a given query, then U_i is the only user who sees the content of that query. Hence it may still be desirable to use Protocols 3 and 5, if additional confidentiality is required.

An interesting related question is, when a particular user U_t sees a query q posted to the memory space S_ℓ that is not his own, whether or not U_t has a probabilistic advantage in guessing the source of q. The following theorems show that, in order to minimize any such advantage, it is helpful to use a regular PBD in our protocols. We begin by considering Protocol 5, the Proxy-designated Covering Design Protocol in which a source never designates himself as proxy in a memory space:

Theorem 3.3. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 5) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Suppose $U_t \in S_\ell$ sees a query q posted to S_ℓ with proxy $U_j \in S_\ell$ that is not his own. Then, from the point of view of U_t, for a given query q and $U_i \in S_\ell$ such that $i \neq t$, it holds that*

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \neq U_t] = \begin{cases} 0 & \text{if } i = j \\ \frac{1}{|S_\ell|-1} & \text{if } t = j \quad (\Rightarrow i \neq j) \\ \frac{1}{|S_\ell|-2} & \text{if } i,\ t \neq j \end{cases}.$$

Proof. We first note that the protocol definition ensures that when $i = j$, we have

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \neq U_t] = 0.$$

We now consider the case $i \neq j$. We set $\lambda_{ij} = |\{S_q : U_i, U_j \in S_q\}| = \lambda$. Thus, we have

$$\begin{aligned} \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \neq U_t] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \Pr[\mathbf{M} = S_\ell \mid \mathbf{P} = U_j, \mathbf{S} = U_i] \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \frac{1}{\lambda_{ij}(v-1)} = \frac{1}{\lambda(v-1)}. \end{aligned}$$

Then, because $i \neq t$, we have

$$\begin{aligned} \Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] &= \frac{\Pr[\mathbf{S} \neq U_t \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i]}{\Pr[\mathbf{S} \neq U_t]} \\ &= \frac{\Pr[\mathbf{S} = U_i]}{\sum_{\substack{U_h \in \mathcal{U} \\ h \neq t}} \Pr[\mathbf{S} = U_h]} \\ &= \frac{1}{v-1}. \end{aligned}$$

This gives

$$\begin{aligned}
&\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \neq U_t] \\
&\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \neq U_t]}{\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} \neq U_t]} \\
&\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i]}{\sum_{\substack{U_h \in S_\ell \\ h \neq t,j}} \Pr[\mathbf{S} = U_h \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_h]} \\
&\quad = \frac{\frac{1}{(v-1)^2\lambda}}{\sum_{\substack{U_h \in S_\ell \\ h \neq t,j}} \frac{1}{(v-1)^2\lambda}} \\
&\quad = \begin{cases} \frac{1}{|S_\ell|-1} & \text{if } t = j \\ \frac{1}{|S_\ell|-2} & \text{if } t \neq j \end{cases},
\end{aligned}$$

as desired. □

We can generalize Theorem 3.3 to a coalition of users C rather than just a single user, as stated in the following theorem. The proof is very similar to the proof of Theorem 3.3.

Theorem 3.4. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 5) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Let C be a coalition of users and suppose (some subset of) C sees a query q posted to S_ℓ with proxy $U_j \in S_\ell$ that was not posted by a member of C. Then, from the point of view of C, for a given query q and $U_i \in S_\ell$ such that $U_i \notin C$, it holds that*

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \notin C] = \begin{cases} 0 & \text{if } i = j \\ \frac{1}{|S_\ell \setminus C|} & \text{if } U_j \in C \quad (\Rightarrow i \neq j) \\ \frac{1}{|S_\ell \setminus C| - 1} & \text{if } U_j \notin C \text{ and } i \neq j \end{cases}.$$

Proof. We first note that the protocol definition ensures that when $i = j$, we have

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \notin C] = 0.$$

We now consider the case $i \neq j$. We set $\lambda_{ij} = |\{S_q : U_i, U_j \in S_q\}| = \lambda$. Thus, we have

$$\begin{aligned}
\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \notin C] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \\
&= \Pr[\mathbf{M} = S_\ell \mid \mathbf{P} = U_j, \mathbf{S} = U_i] \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \\
&= \frac{1}{\lambda_{ij}(v-1)} = \frac{1}{\lambda(v-1)}. \qquad (12)
\end{aligned}$$

Then because $U_i \notin C$, we have

$$\begin{aligned}\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C] &= \frac{\Pr[\mathbf{S} \notin C \mid \mathbf{S} = U_i]\Pr[\mathbf{S} = U_i]}{\Pr[\mathbf{S} \notin C]} \\ &= \frac{\Pr[\mathbf{S} = U_i]}{\sum_{U_h \in \mathcal{U}\setminus C}\Pr[\mathbf{S} = U_h]} \\ &= \frac{1}{v - |C|}.\end{aligned}$$

This gives

$$\begin{aligned}&\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \notin C] \\ &\quad= \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C]\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \notin C]}{\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} \notin C]} \\ &\quad= \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C]\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i]}{\sum_{\substack{U_h \in S_\ell\setminus C \\ h \neq j}}\Pr[\mathbf{S} = U_h \mid \mathbf{S} \notin C]\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_h]} \\ &\quad= \frac{\frac{1}{(v-|C|)(v-1)\lambda}}{\sum_{\substack{U_h \in S_\ell\setminus C \\ h \neq j}}\frac{1}{(v-|C|)(v-1)\lambda}} \\ &\quad= \begin{cases} \frac{1}{|S_\ell\setminus C|} & \text{if } U_j \in C \\ \frac{1}{|S_\ell\setminus C|-1} & \text{if } U_j \notin C \end{cases},\end{aligned}$$

as desired. □

Theorem 3.3 implies that for $U_t \in S_\ell$, if U_t sees a query q with proxy U_j posted to the memory space S_ℓ that is not his own, any of the remaining $|S_\ell\setminus\{U_j, U_t\}|$ users in S_ℓ are equally likely to be the source. Similarly, in Theorem 3.4, we observe that from the point of view of a coalition C, the set of possible sources is $S_\ell\setminus(C \cup U_j)$, and of these possibilities, any user is equally likely.

We now give the parallel results for Protocol 6, the Proxy-designated Covering Design Protocol, in which a source is allowed to act as his own proxy. In this case, a single user U_t (or a coalition C) can no longer completely eliminate the possibility of the proxy U_j being the source. However, as the following theorems show, the likelihood of the proxy U_j being the source is not the same as the likelihood of $U_i \neq U_j$ being the source. Indeed, it is far less likely that U_j is acting as both proxy and source for q in this situation. Intuitively, if a user U_i is acting as both source and proxy, he has r possible memory spaces to choose from, whereas if U_i chooses another user U_j as proxy, he has only λ many memory spaces to choose from.

Theorem 3.5. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 6) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$*

for all $U_i \in \mathcal{U}$. Suppose $U_t \in S_\ell$ sees a query q posted to S_ℓ with proxy $U_j \in S_\ell$ that is not his own. Then, from the point of view of U_t, for a given query q and $U_i \in S_\ell$ such that $i \neq t$, it holds that

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \neq U_t] = \begin{cases} \frac{1}{|S_\ell|-1} & \text{if } t = j \quad (\Rightarrow i \neq j) \\ \frac{\lambda}{\lambda+r(|S_\ell|-2)} & \text{if } i = j \quad (\Rightarrow t \neq j) \\ \frac{r}{\lambda+r(|S_\ell|-2)} & \text{if } i \neq j,\ t \neq j \end{cases}.$$

Proof. We first calculate $\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \neq U_t]$, where $U_i \in S_\ell$.

We again set $\lambda_{ij} = |\{S_q : U_i, U_j \in S_q\}|$. In particular, since (X, A) is a PBD of degree r, we have

$$\lambda_{ij} = \begin{cases} r & \text{if } i = j \\ \lambda & \text{if } i \neq j \end{cases}.$$

We have

$$\begin{aligned} \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \neq U_t] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \Pr[\mathbf{M} = S_\ell \mid \mathbf{P} = U_j, \mathbf{S} = U_i] \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \left(\frac{1}{\lambda_{ij}}\right)\left(\frac{1}{v}\right). \end{aligned} \tag{13}$$

Then we have $\Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] = \frac{1}{v-1}$ and this gives

$$\begin{aligned} &\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \neq U_t] \\ &\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \neq U_t]}{\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} \neq U_t]} \\ &\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i]}{\sum_{\substack{U_h \in S_\ell \\ h \neq t}} \Pr[\mathbf{S} = U_h \mid \mathbf{S} \neq U_t] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_h]} \\ &\quad = \frac{\frac{1}{v(v-1)\lambda_{ij}}}{\sum_{\substack{U_h \in S_\ell \\ h \neq t}} \frac{1}{v(v-1)\lambda_{hj}}} \\ &\quad = \begin{cases} \frac{\lambda}{\lambda_{ij}(|S_\ell|-1)} & \text{if } t = j \\ \frac{\lambda r}{\lambda_{ij}\left(\lambda+r(|S_\ell|-2)\right)} & \text{if } t \neq j \end{cases}, \end{aligned}$$

which yields the desired result. □

Theorem 3.5 also generalizes nicely to coalitions:

Theorem 3.6. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 6) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Let C be a coalition of users and suppose (some subset of) C sees a query*

q posted to S_ℓ with proxy $U_j \in S_\ell$ that was not posted by a member of C. Then, from the point of view of C, for a given query q and $U_i \in S_\ell$ such that $U_i \notin C$, it holds that

$$\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \notin C] = \begin{cases} \frac{1}{|S_\ell \setminus C|} & \text{if } U_j \in C \quad (\Rightarrow i \neq j) \\ \frac{\lambda}{\lambda + r(|S_\ell \setminus C| - 1)} & \text{if } U_j \notin C,\ i = j \\ \frac{r}{\lambda + r(|S_\ell \setminus C| - 1)} & \text{if } U_j \notin C,\ i \neq j \end{cases}.$$

Proof. We first calculate $\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \notin C]$, where $U_i \in S_\ell$. We again set $\lambda_{ij} = |\{S_q : U_i, U_j \in S_q\}|$. Since (X, A) is a PBD of degree r, we have

$$\lambda_{ij} = \begin{cases} r & \text{if } i = j \\ \lambda & \text{if } i \neq j \end{cases}.$$

We have

$$\begin{aligned} \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \notin C] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \Pr[\mathbf{M} = S_\ell \mid \mathbf{P} = U_j, \mathbf{S} = U_i] \Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i] \\ &= \left(\frac{1}{\lambda_{ij}}\right)\left(\frac{1}{v}\right). \end{aligned}$$

Then we have $\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C] = \frac{1}{v - |C|}$ and this gives

$$\begin{aligned} &\Pr[\mathbf{S} = U_i \mid \mathbf{M} = S_\ell, \mathbf{P} = U_j, \mathbf{S} \notin C] \\ &\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{S} \notin C]}{\Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} \notin C]} \\ &\quad = \frac{\Pr[\mathbf{S} = U_i \mid \mathbf{S} \notin C] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i]}{\sum_{U_h \in S_\ell \setminus C} \Pr[\mathbf{S} = U_h \mid \mathbf{S} \notin C] \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_h]} \\ &\quad = \frac{\frac{1}{v(v - |C|)\lambda_{ij}}}{\sum_{U_h \in S_\ell \setminus C} \frac{1}{v(v - |C|)\lambda_{hj}}} \\ &\quad = \frac{1}{\lambda_{ij} \sum_{U_h \in S_\ell \setminus C} \lambda_{hj}} \\ &\quad = \begin{cases} \frac{\lambda}{\lambda_{ij} |S_\ell \setminus C|} & \text{if } U_j \in C \\ \frac{\lambda r}{\lambda_{ij}\left(\lambda + r(|S_\ell \setminus C| - 1)\right)} & \text{if } U_j \notin C \end{cases}, \end{aligned}$$

which yields the desired result. □

Remark 3.9. Theorems 3.3–3.4 apply to Protocol 3 and Theorems 3.5–3.6 apply to Protocol 4. This follows immediately, since a BIBD is also a regular PBD of degree r.

We observe that Theorems 3.3–3.4 and Theorems 3.5–3.6 demonstrate that the use of a regular PBD increases privacy against other users. This is because, from the point of view of another user U_t (or coalition of users C), only three possible probabilities arise in

the conditional source distribution if a regular PBD is used, thereby preventing U_t (or C) from being able to distinguish between users within each of these sets. In particular, if a regular PBD is not used, the parameter λ_{ij} is not a constant determined by whether $i = j$ or not, but instead may vary across all pairs U_i, $U_j \in \mathcal{U}$.

3.6.1. Linked queries and coalitions of users. Users can also launch an intersection attack against a series of linked queries, similar to the intersection attack launched by DB against the DBWM and DBWMS protocols (Protocols 1 and 2). The difference here is that users have access to the content of queries via the shared memory spaces; that is, users of a given memory space know which queries have been posted to that memory space, whereas the database only knows the identity of the proxy. In particular, since more information is available to users than to the database, it is correspondingly more difficult to provide privacy in the presence of linked queries against coalitions of users, than against the database.

EXAMPLE 3.4. Consider the projective plane from Example 3.2 and suppose we use Protocol 4. Suppose that U_1 is the source of two linked queries, where the first query uses memory space $\{U_1, U_2, U_3\}$ and the second query uses memory space $\{U_1, U_4, U_5\}$. Now suppose that users U_2 and U_5 collude. From the first query, user U_2 knows that the source $U_i \in \{U_1, U_3\}$ (regardless of the proxy). From the second query, user U_5 knows that the source $U_i \in \{U_1, U_4\}$ (regardless of the proxy). If users U_2 and U_5 collude, then they can identify U_1 as the source.

In general, we can consider a sequence of ρ linked queries made by the same (unknown) user, and a coalition C of at most c users that is trying to identify the source of the ρ queries. We introduce the following terminology.

Definition 3.3. Consider a set of ρ linked queries and fix a maximum coalition size c. If there are always at least κ users who could possibly be the source (regardless of the queries and coalition) then we say that the scheme provides (ρ, c, κ)-*anonymity*.

REMARK 3.10. Of course we want $\kappa \geq 2$ because the source might be identified if $\kappa = 1$.

In general, analyzing the security of our protocols against coalition attacks on linked queries is difficult and depends on the block intersection properties of the underlying design. Moreover, the notion of (ρ, c, κ)-anonymity does not take into account the probabilistic advantage a coalition C has in guessing the source of a series of linked queries based on the memory spaces used and the identities of the proxies. In this section, we consider some special cases in which it is easy to determine the level of (ρ, c, κ)-anonymity provided and discuss the advantage coalitions have in determining the source of a series of linked queries.

First, however, we make some general observations about our protocols with respect to the expected distribution of proxies. These observations lead to an attack similar to the projective plane attack [**73**] mentioned in Section 3.4.1 and the *predecessor attack* on

Crowds [56, 82, 83], which we discuss in Section 3.7. In particular, the source U_i of a series of linked queries acts as his own proxy fewer times than any other user $U_j \in \mathcal{U} \backslash \{U_i\}$ over the course of those linked queries *on average*. Therefore, a coalition C of users can, given sufficiently many linked queries, identify the source with some probability that approaches 1. The number of linked queries necessary to successfully perform this attack (with some appropriately small error probability) can be estimated using Chernoff bounds [52], if desired. This attack is an obvious consequence of the following theorem:

Theorem 3.7. *Assume we use a regular PBD of degree r in Protocol 5 or 6 and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Consider a user U_i and a memory space S_ℓ satisfying $U_i \in S_\ell$. Let $q_1, \ldots, q_\rho$ be a series of queries with source U_i posted to S_ℓ. In Protocol 5, the user U_i never designates himself as proxy in a memory space, whereas every other user $U_j \neq U_i \in S_\ell$ acts as a proxy for U_i an average of $\rho\left(\frac{r}{\lambda(v-1)}\right)$ times over the course of the ρ queries. In Protocol 6, U_i acts as his own proxy an average of $\frac{\rho}{v}$ times, and every other user $U_j \neq U_i \in S_\ell$ acts as proxy an average of $\rho\left(\frac{r}{\lambda v}\right)$ times.*

Proof. We assume $U_j \in S_\ell$.

For both Protocols 5 and 6, we have

$$\begin{aligned}\Pr[\mathbf{S} = U_i, \mathbf{M} = S_\ell] &= \Pr[\mathbf{M} = S_\ell \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i] \\ &= \frac{1}{rv}.\end{aligned}$$

Now we consider Protocol 5. For $i \neq j$, we have

$$\begin{aligned}\Pr[\mathbf{P} = U_j, \mathbf{S} = U_i, \mathbf{M} = S_\ell] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i] \\ &= \frac{1}{\lambda v(v-1)} \text{ (by Equation (12)).}\end{aligned}$$

This gives

$$\begin{aligned}\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{M} = S_\ell] &= \frac{\Pr[\mathbf{P} = U_j, \mathbf{S} = U_i, \mathbf{M} = S_\ell]}{\Pr[\mathbf{S} = U_i, \mathbf{M} = S_\ell]} \\ &= \frac{\left(\frac{1}{\lambda v(v-1)}\right)}{\left(\frac{1}{rv}\right)} \\ &= \frac{r}{\lambda(v-1)}.\end{aligned}$$

This gives the desired result for Protocol 5.

For Protocol 6, we have

$$\begin{aligned}\Pr[\mathbf{P} = U_j, \mathbf{S} = U_i, \mathbf{M} = S_\ell] &= \Pr[\mathbf{M} = S_\ell, \mathbf{P} = U_j \mid \mathbf{S} = U_i] \Pr[\mathbf{S} = U_i] \\ &= \frac{1}{\lambda_{ij} v^2} \text{ (by Equation (13)).}\end{aligned}$$

This gives

$$\begin{aligned}\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{M} = S_\ell] &= \frac{\Pr[\mathbf{P} = U_j, \mathbf{S} = U_i, \mathbf{M} = S_\ell]}{\Pr[\mathbf{S} = U_i, \mathbf{M} = S_\ell]} \\ &= \frac{\left(\frac{1}{\lambda_{ij} v^2}\right)}{\left(\frac{1}{rv}\right)} \\ &= \frac{r}{\lambda_{ij} v} \\ &= \begin{cases} 1/v & \text{if } i = j \\ r/\lambda v & \text{if } i \neq j \end{cases}.\end{aligned}$$

This gives the desired result for Protocol 6. □

We now consider "coalitions" consisting of a single user (i.e., the case $c = 1$). For anonymity against other users, it is advantageous to use a BIBD with $\lambda = 1$:

Lemma 3.8. *Suppose the BIBD chosen for Protocol 4 satisfies $\lambda = 1$. Then we achieve $(\rho, 1, k-1)$-anonymity for any ρ.*

Proof. Suppose U_i sees a sequence of ρ linked queries from the same source. Then in particular, the queries must all involve the same memory space, because $\lambda = 1$. The result then follows from Theorem 3.5. □

On the other hand, the security of Protocol 4 against a single user might be completely eliminated if we use a design with $\lambda > 1$. For example, suppose we use a BIBD with $\lambda = 2$ in which every pair of blocks intersects in at most two points (i.e., a *supersimple* BIBD, as defined in Definition 1.20). Consider two users U_i and U_j. There exist two memory spaces, say S_1 and S_2, where $S_1 \cap S_2 = \{U_i, U_j\}$. Suppose U_i observes two linked queries, say q_1 and q_2, that involve S_1 and S_2, respectively. Then U_i can deduce that U_j is the source.

Remark 3.11. The result of Lemma 3.8 does not apply to Protocol 3. This is because in Protocol 3, given a series of linked queries posted to a given memory space, the only user who never acts as proxy for one of these queries is the query issuer, as discussed in Theorem 3.7.

Remark 3.12. Lemma 3.8 states that after observing a series of ρ linked queries, a user U_t can narrow down the set of possible sources to $k-1$ other users. This is not to say that U_t has no advantage in guessing the source. In particular, as Theorem 3.7 indicates, the source is probabilistically under-represented as proxy. That is, given a large enough ρ, the user U_t has the ability to distinguish the source based on the observed proxy distribution.

We now analyze (ρ, c, κ)-anonymity for small values of ρ. Here it is helpful to consider certain types of designs that have useful properties relating to block intersections. In

Section 3.6.2, we consider a more general approach to mitigate intersection attacks in the Proxy-designated Covering Design Protocols (Protocols 5 and 6).

The case $\rho = 1$ (i.e., security against a single query) is easy to analyze:

Lemma 3.9. *We achieve $(1, c, k-c-1)$-anonymity in Protocol 3, where $c \leq k-3$. In Protocol 4, we achieve $(1, c, k-c)$-anonymity, with the requirement that $c \leq k-2$.*

Proof. We first consider Protocol 3. Let C be a coalition of size at most c and let S_h be the memory space used for the query q_1. Then $|C \cap S_h| \leq c$. C can rule out as possible sources the users in $C \cap S_h$ as well as the proxy U_j (provided that $U_j \notin C \cap S_h$). Since $|S_h \backslash (C \cup \{U_j\})| \geq k-c-1$, the result follows. An obvious requirement here is $c \leq k-3$.

For Protocol 4, all other users with access to the given memory space can only eliminate themselves as the possible source of the query. This improves the information-theoretic security for user privacy against other users, as we now have $|S_h \backslash C| \geq k-c$. An obvious requirement here is $c \leq k-2$. □

Moreover, Theorems 3.4 and 3.6 give the advantage a coalition C of size c has in guessing the identity of the actual source from the set of possible sources.

For the case $\rho = 2$, it is helpful to consider BIBDs with a special intersection property.

Lemma 3.10. *Suppose the BIBD of Protocols 3 and 4 satisfies the additional property that any two blocks intersect in at least μ points. Consider two linked queries, q_1 and q_2. Then we achieve $(2, c, \mu-c-2)$-anonymity, where $c \leq \mu-4$, in Protocol 3. In Protocol 4, we achieve $(2, c, \mu-c)$-anonymity, with the requirement $c \leq \mu-2$.*

Proof. Let C be a coalition of size at most c and let S_{h_1} be the memory space used for the query q_1 and S_{h_2} be the memory space used for q_2. Let U_i be the proxy for q_1 and let U_j be the proxy for q_2.

In Protocol 3, we have

$$\left|\left(S_{h_1} \setminus (C \cup \{U_i\})\right) \cap \left(S_{h_2} \setminus (C \cup \{U_j\})\right)\right| = \left|(S_{h_1} \cap S_{h_2}) \setminus (C \cup \{U_i, U_j\})\right| \geq \mu - c - 2,$$

so we achieve $(2, c, \mu-c-2)$-anonymity. An obvious requirement here is $c \leq \mu-4$.

In Protocol 4, we have

$$\left|(S_{h_1} \backslash C) \cap (S_{h_2} \backslash C)\right| = \left|(S_{h_1} \cap S_{h_2}) \backslash C\right| \geq \mu - c,$$

so we achieve $(2, c, \mu-c)$-anonymity. Here, an obvious requirement is $c \leq \mu-2$. □

We can apply Lemma 3.10 to the case of a symmetric BIBD, in which any two blocks intersect in exactly λ points, as noted in Theorem 1.4. This achieves the following result:

Corollary 3.11. *Suppose the BIBD chosen for Protocol 3 or 4 is a symmetric (v, v, k, k, λ)-BIBD. Then Protocol 3 provides $(2, c, \lambda-c-2)$-anonymity for any $c \leq \lambda-4$ and Protocol 4 provides $(2, c, \lambda-c)$-anonymity for any $c \leq \lambda-2$.*

We can derive the advantage a coalition C has in guessing the source of a series of two linked queries q_1 and q_2. This can be generalized to ρ linked queries in the obvious way, but in general the number of possible sources is likely to decrease quickly when more than one or two memory spaces are involved. We state our results in Theorems 3.12 and 3.13; these results can be applied to Lemma 3.10 and Corollary 3.11 in order to determine the coalition C's advantage in guessing the source of two linked queries.

First, we establish the following useful notation to discuss two linked queries q_1 and q_2, which we use in both Theorems 3.12 and 3.13. In both, we assume that there are two linked queries q_1 and q_2, where query q_1 is posted to S_{ℓ_1} with proxy $U_{j_1} \in S_{\ell_1}$ and query q_2 is posted to S_{ℓ_2} with proxy $U_{j_2} \in S_{\ell_2}$. We use random variables $\mathbf{S_1}$, $\mathbf{M_1}$, and $\mathbf{P_1}$ to denote the source, memory space, and proxy with respect to query q_1 and random variables $\mathbf{S_2}$, $\mathbf{M_2}$, and $\mathbf{P_2}$ to denote the source, memory space, and proxy with respect to query q_2. In addition, we define the following events:

1. Let Q_1 denote the event $\mathbf{M_1} = S_{\ell_1}$ and $\mathbf{P_1} = U_{j_1}$.
2. Let Q_2 denote the event $\mathbf{M_2} = S_{\ell_2}$ and $\mathbf{P_2} = U_{j_2}$.
3. For $U_h \in \mathcal{U}$, let E_h denote the event $\mathbf{S_1} = \mathbf{S_2} = U_h$.

Theorem 3.12. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 5) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Let C be a coalition of users and suppose (some subset of) C sees two linked queries q_1 and q_2 that were not posted by a member of C, where q_1 is posted to S_{ℓ_1} with proxy $U_{j_1} \in S_{\ell_1}$ and q_2 is posted to S_{ℓ_2} with proxy $U_{j_2} \in S_{\ell_2}$. Then, from the point of view of C, for $U_i \notin C$, it holds that*

1. *If $U_i \notin S_{\ell_1} \cap S_{\ell_2}$ or if $i \in \{j_1, j_2\}$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = 0.$$
2. *If $U_i \in S_{\ell_1} \cap S_{\ell_2}$ and $i \notin \{j_1, j_2\}$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = \frac{1}{\left|\left(S_{\ell_1} \cap S_{\ell_2}\right) \setminus \left(C \cup \{U_{j_1}, U_{j_2}\}\right)\right|}.$$

Proof. We proceed as in the proof of Theorem 3.4. As before, a user U_i cannot be the source for a given query if he is also the proxy, so we restrict ourselves to the case of users $U_i \notin C$ satisfying $i \notin \{j_1, j_2\}$.

We then calculate

$$\begin{aligned}
\Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_i] &= \Pr[Q_1, Q_2 \mid \mathbf{S_1}, \mathbf{S_2} \notin C; E_i] \\
&= \Pr[Q_1 \mid \mathbf{S_1} \notin C, \mathbf{S_1} = U_i] \Pr[Q_2 \mid \mathbf{S_2} \notin C, \mathbf{S_2} = U_i] \\
&= \begin{cases} 0 & \text{if } U_i \notin S_{\ell_1} \cap S_{\ell_2} \\ \left(\frac{1}{\lambda(v-1)}\right)^2 & \text{otherwise} \end{cases}.
\end{aligned}$$

Note that the above computation implies, as expected, that if $U_i \notin S_{\ell_1} \cap S_{\ell_2}$,

$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = 0.$$

For the case $U_i \in S_{\ell_1} \cap S_{\ell_2}$, we have

$$\begin{aligned}
\Pr[\mathbf{S_1} = \mathbf{S_2} \notin C] &= \sum_{U_h \notin C} \Pr[E_h] \\
&= \sum_{U_h \notin C} \Pr[\mathbf{S_1} = U_h] \Pr[\mathbf{S_2} = U_h] \\
&= \sum_{U_h \notin C} \frac{1}{v^2} \\
&= \frac{v - |C|}{v^2}.
\end{aligned}$$

This last computation allows us to determine

$$\begin{aligned}
\Pr[E_i \mid \mathbf{S_1} = \mathbf{S_2} \notin C] &= \frac{\Pr[\mathbf{S_1} = \mathbf{S_2} \notin C \mid E_i] \Pr[E_i]}{\Pr[\mathbf{S_1} = \mathbf{S_2} \notin C]} \\
&= \frac{\Pr[E_i]}{\Pr[\mathbf{S_1} = \mathbf{S_2}; \mathbf{S_1}, \mathbf{S_2} \notin C]} \\
&= \frac{1}{v - |C|}.
\end{aligned}$$

Finally, for $U_i \notin C$ satisfying $U_i \in S_{\ell_1} \cap S_{\ell_2}$, we have

$$\begin{aligned}
&\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] \\
&\quad = \frac{\Pr[E_i \mid \mathbf{S_1} = \mathbf{S_2} \notin C] \Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_i]}{\Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C]} \\
&\quad = \frac{\left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda(v-1)}\right)^2}{\sum_{\substack{U_h \in (S_{\ell_1} \cap S_{\ell_2}) \setminus C \\ h \neq j_1, j_2}} \Pr[E_h \mid \mathbf{S_1} = \mathbf{S_2} \notin C] \Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_h]} \\
&\quad = \frac{\left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda(v-1)}\right)^2}{\sum_{\substack{U_h \in (S_{\ell_1} \cap S_{\ell_2}) \setminus C \\ h \neq j_1, j_2}} \left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda(v-1)}\right)^2} \\
&\quad = \frac{1}{\left|\left(S_{\ell_1} \cap S_{\ell_2}\right) \setminus \left(C \cup \{U_{j_1}, U_{j_2}\}\right)\right|}.
\end{aligned}$$

This completes the proof. □

Theorem 3.13. *Let (X, A) be a regular PBD of degree r. Assume (X, A) is used in the Proxy-designated Covering Design Protocol (Protocol 6) and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Let C be a coalition of users and suppose (some subset of) C sees two linked queries q_1 and q_2 that were not posted by a member of C, where q_1 is posted to S_{ℓ_1} with proxy $U_{j_1} \in S_{\ell_1}$ and q_2 is posted to S_{ℓ_2} with proxy $U_{j_2} \in S_{\ell_2}$. Then, from the point of view of C, for $U_i \notin C$, it holds that*

1. *If $U_i \notin S_{\ell_1} \cap S_{\ell_2}$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = 0.$$
2. *If $U_i \in S_{\ell_1} \cap S_{\ell_2}$ and U_{j_1}, $U_{j_2} \notin (S_{\ell_1} \cap S_{\ell_2}) \backslash C$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = \frac{1}{\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right|}.$$
3. *If $U_i \in S_{\ell_1} \cap S_{\ell_2}$ and precisely one of U_{j_1}, $U_{j_2} \in (S_{\ell_1} \cap S_{\ell_2}) \backslash C$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = \begin{cases} \frac{\lambda}{\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 1\right)} & \text{if } i \in \{j_1, j_2\} \\ \frac{r}{\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 1\right)} & \text{if } i \notin \{j_1, j_2\} \end{cases}.$$
4. *If $U_i \in S_{\ell_1} \cap S_{\ell_2}$ and U_{j_1}, $U_{j_2} \in (S_{\ell_1} \cap S_{\ell_2}) \backslash C$, then*
$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = \begin{cases} \frac{\lambda^2}{\lambda^2 + r^2\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 1\right)} & \text{if } i = j_1 = j_2 \\ \frac{\lambda}{2\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 2\right)} & \text{if } i \in \{j_1, j_2\},\ j_1 \neq j_2 \\ \frac{r}{2\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 2\right)} & \text{if } i \notin \{j_1, j_2\},\ j_1 \neq j_2 \\ \frac{r^2}{\lambda^2 + r^2\left(\left|(S_{\ell_1} \cap S_{\ell_2}) \backslash C\right| - 1\right)} & \text{if } i \notin \{j_1, j_2\},\ j_1 = j_2 \end{cases}.$$

Proof. We proceed as in the proof of Theorem 3.6. We first calculate
$$\Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_i],$$
where $U_i \notin C$. We again set $\lambda_{ij} = |\{S_q : U_i, U_j \in S_q\}|$. Since (X, A) is a PBD of degree r, we have
$$\lambda_{ij} = \begin{cases} r & \text{if } i = j \\ \lambda & \text{if } i \neq j \end{cases}.$$

We then calculate
$$\begin{aligned} \Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_i] &= \Pr[Q_1, Q_2 \mid \mathbf{S_1}, \mathbf{S_2} \notin C; E_i] \\ &= \Pr[Q_1 \mid \mathbf{S_1} \notin C, \mathbf{S_1} = U_i] \Pr[Q_2 \mid \mathbf{S_2} \notin C, \mathbf{S_2} = U_i] \\ &= \begin{cases} 0 & \text{if } U_i \notin S_{\ell_1} \cap S_{\ell_2} \\ \frac{1}{\lambda_{ij_1} \lambda_{ij_2} v^2} & \text{otherwise} \end{cases}. \end{aligned}$$

Note that the above computation implies, as expected, that if $U_i \notin S_{\ell_1} \cap S_{\ell_2}$

$$\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] = 0.$$

For the case $U_i \in S_{\ell_1} \cap S_{\ell_2}$, as in the proof of Theorem 3.12, we have

$$\Pr[\mathbf{S_1} = \mathbf{S_2} \notin C] = \frac{v - |C|}{v^2} \quad \text{and} \quad \Pr[E_i \mid \mathbf{S_1} = \mathbf{S_2} \notin C] = \frac{1}{v - |C|}.$$

Finally, for $U_i \notin C$ satisfying $U_i \in S_{\ell_1} \cap S_{\ell_2}$, we have

$$\begin{aligned}
&\Pr[E_i \mid Q_1, Q_2; \mathbf{S_1} = \mathbf{S_2} \notin C] \\
&\quad = \frac{\Pr[E_i \mid \mathbf{S_1} = \mathbf{S_2} \notin C] \Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_i]}{\Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C]} \\
&\quad = \frac{\left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda_{ij_1}\lambda_{ij_2}v^2}\right)}{\sum_{U_h \in (S_{\ell_1} \cap S_{\ell_2})\setminus C} \Pr[E_h \mid \mathbf{S_1} = \mathbf{S_2} \notin C] \Pr[Q_1, Q_2 \mid \mathbf{S_1} = \mathbf{S_2} \notin C; E_h]} \\
&\quad = \frac{\left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda_{ij_1}\lambda_{ij_2}v^2}\right)}{\sum_{U_h \in (S_{\ell_1} \cap S_{\ell_2})\setminus C} \left(\frac{1}{v-|C|}\right)\left(\frac{1}{\lambda_{hj_1}\lambda_{hj_2}v^2}\right)} \\
&\quad = \frac{1}{\lambda_{ij_1}\lambda_{ij_2} \sum_{U_h \in (S_{\ell_1} \cap S_{\ell_2})\setminus C} \left(\frac{1}{\lambda_{hj_1}\lambda_{hj_2}}\right)} \\
&\quad = \begin{cases}
\frac{\lambda^2}{\lambda_{ij_1}\lambda_{ij_2}\left|(S_{\ell_1} \cap S_{\ell_2})\setminus C\right|} & \text{if } U_{j_1}, U_{j_2} \notin (S_{\ell_1} \cap S_{\ell_2})\setminus C \\
\frac{\lambda^2 r}{\lambda_{ij_1}\lambda_{ij_2}\left(\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2})\setminus C\right| - 1\right)\right)} & \text{if precisely one of } U_{j_1}, U_{j_2} \in (S_{\ell_1} \cap S_{\ell_2})\setminus C \\
\frac{\lambda^2 r}{\lambda_{ij_1}\lambda_{ij_2}\left(2\lambda + r\left(\left|(S_{\ell_1} \cap S_{\ell_2})\setminus C\right| - 2\right)\right)} & \text{if } U_{j_1}, U_{j_2} \in (S_{\ell_1} \cap S_{\ell_2})\setminus C,\ j_1 \neq j_2 \\
\frac{\lambda^2 r^2}{\lambda_{ij_1}\lambda_{ij_2}\left(\lambda^2 + r^2\left(\left|(S_{\ell_1} \cap S_{\ell_2})\setminus C\right| - 1\right)\right)} & \text{if } U_{j_1}, U_{j_2} \in (S_{\ell_1} \cap S_{\ell_2})\setminus C,\ j_1 = j_2
\end{cases}.
\end{aligned}$$

This completes the proof. □

Another extension to the concept of (ρ, c, κ)-anonymity is to consider an average-case analysis of privacy against other users. Thus far, we have analyzed the worst-case scenario—the minimum level of privacy the scheme achieves against *any* possible coalition. While this is useful in some respects, schemes suffering powerful worst-case scenario attacks might actually perform quite well against a typical (i.e., random) coalition. In particular, if a scheme needs to be concerned about random coalitions of users, such an average-case analysis might prove informative, as the following example shows.

Example 3.5. Suppose we use a symmetric $(v, v, k, k, 3)$-BIBD in Protocol 6. Consider linked queries q_1 and q_2 submitted by U_i, with corresponding memory spaces S_{h_1} and S_{h_2}. By Theorem 1.4, since the BIBD is symmetric, we have $|S_{h_1} \cap S_{h_2}| = 3$. That is, there are

exactly two other users, say U_j and U_t, in both S_{h_1} and S_{h_2}. This implies that there is only *one* coalition of users of size 2 that can identify U_i as the source. If we consider random coalitions, the probability that a random coalition of size 2 consists of $\{U_j, U_t\}$ is $1/\binom{v-1}{2}$.

Let us consider other coalitions of size 2. Suppose $C = \{U_j, U_\ell\}$, for some user $U_\ell \neq U_t, U_i$. Then C knows the source is either U_t or U_i. There are $v-3$ such coalitions. The analysis for C containing U_t but not U_j is similar. If we consider $C = \{U_\ell, U_{\ell'}\}$ such that $U_t, U_j \notin C$, the most advantageous coalition satisfies (without loss of generality) $U_\ell \in S_{h_1}$, $U_{\ell'} \in S_{h_2}$. In this case, C sees both q_1 and q_2 and can conclude that the source is one of $\{U_i, U_j, U_t\}$. There are $(k-3)^2$ such coalitions. Other coalitions of size 2 either see only one of $\{q_1, q_2\}$, in which case the analysis reduces to that of Theorem 3.4 or 3.6, or neither of the linked queries, in which case C can do nothing.

We extend this average-case analysis for $\rho = 2$ to a general symmetric (v, k, λ)-BIBD and coalitions C of size c in the following example. The analysis provided can be used to compute the performance of Protocol 6 against an average coalition when a symmetric BIBD is used.

Theorem 3.14. *Suppose we use a symmetric (v, k, λ)-BIBD in Protocol 6. Consider linked queries q_1 and q_2 submitted by U_i, with corresponding memory spaces S_{h_1} and S_{h_2}. Suppose we have a coalition C of size c. We have the following breakdown of possible coalitions C that can occur:*

1. *There are a total of $2\binom{v-k}{c} - \binom{v-2k+\lambda}{c}$ coalitions C that do not see both q_1 and q_2. In particular, there are*
   (a) *$\binom{v-2k+\lambda}{c}$ coalitions such that C sees neither of the queries, and*
   (b) *$2\left(\binom{v-k}{c} - \binom{v-2k+\lambda}{c}\right)$ coalitions such that C sees only one of the queries.*
2. *The remaining coalitions C see both q_1 and q_2. In particular, there are*
   (a) *$\binom{\lambda-1}{t}\binom{v-\lambda}{c-t}$ coalitions such that C has exactly t members in $S_{h_1} \cap S_{h_2}$, where here t satisfies $1 \leq t \leq \lambda - 1$, and*
   (b) *$\binom{v-\lambda}{c} - 2\binom{v-k}{c} + \binom{v-2k+\lambda}{c}$ coalitions such that $C \cap (S_{h_1} \cap S_{h_2}) = \emptyset$, but $C \cap S_{h_1} \neq \emptyset$ and $C \cap S_{h_2} \neq \emptyset$.*

Now, if we have a symmetric BIBD, we have $|S_{h_1} \cap S_{h_2}| = \lambda$, by Theorem 1.4. Considering the breakdown of coalitions C given in Theorem 3.14, we make the following observations about the capabilities of each of these coalitions with respect to linked queries q_1 and q_2. First, coalitions C that cannot view either of the queries are unable to launch an attack in the first place. For coalitions that see exactly one of the queries, the analysis reduces to that of a single query. In particular, such a coalition knows the source is one of the $|S_{h_j} \backslash C|$ other users in the given memory space S_{h_j}. We can use Theorem 3.6 to determine C's advantage in guessing the identity of the source.

Now suppose the coalition can view both q_1 and q_2. In all such cases, C can determine that $U_i \in (S_{h_1} \cap S_{h_2}) \backslash C$. Theorem 3.13 can be used to determine the coalition's advantage

in guessing the source U_i. In particular, for coalitions that have exactly t members in $S_{h_1} \cap S_{h_2}$, where t satisfies $1 \leq t \leq \lambda - 1$, we have $|(S_{h_1} \cap S_{h_2}) \backslash C| = \lambda - t$. The only other possible type of coalition C that can view q_1 and q_2 has no member in both S_{h_1} and S_{h_2}, but has at least one member in each of these memory spaces. For this type of coalition, we have $|(S_{h_1} \cap S_{h_2}) \backslash C| = \lambda$.

EXAMPLE 3.6. Consider the finite projective plane of order 17, which is a symmetric $(307, 18, 1)$-BIBD. Consider linked queries q_1 and q_2 submitted by U_i, with corresponding memory spaces S_{h_1} and S_{h_2}. Using the analysis in Example 3.14, we consider security against a coalition of size 2. We have the following breakdown of coalitions:

- There are $\binom{v-2k+\lambda}{c} = 36856$ coalitions that see neither q_1 nor q_2.
- There are $2\left(\binom{v-k}{2} - \binom{v-2k+\lambda}{2}\right) = 9520$ coalitions that see only one query $q_j \in \{q_1, q_2\}$. These coalitions are able to narrow down the list of possible sources to $S_{h_j} \backslash C$. We note that $|S_{h_j} \backslash C| \geq k - 2 = 16$.
- There are $\binom{v-\lambda}{c} - 2\binom{v-k}{c} + \binom{v-2k+\lambda}{c} = (k-\lambda)^2 = 289$ coalitions that see both q_1 and q_2. These coalitions can identify the source U_i.

The probability that a random coalition of size 2 can identify the source is therefore approximately 0.006.

3.6.2. Two methods to increase privacy. We discuss two methods for increasing privacy against coalitions of users, namely incorporating *t-anonymity sets* and *query hops* into our protocols.

3.6.2.1. *t-anonymity sets.* Beyond the limited cases described above, it is difficult to analyze the privacy guarantees of the proxy-designated BIBD and covering design protocols in the presence of linked queries. In particular, it becomes difficult to analyze the case of intersections of three or more memory spaces, and the size of these intersections probably decreases quickly. We might, however, wish to provide privacy for $\rho > 2$. One possible solution is to introduce the notion of built-in *permanent anonymity sets* for each user. That is, suppose the set of users $\mathcal{U}$ is partitioned into *anonymity sets* $\mathcal{T}_1, \ldots \mathcal{T}_g$, where each $\mathcal{T}_\ell$ consists of at least t users. We further assume that the set system satisfies the property $\mathcal{T}_\ell \cap S_j \in \{\emptyset, \mathcal{T}_\ell\}$ for all ℓ, j. We call such a construction a *covering design with t-anonymity sets.*

Theorem 3.15. *Fix a partition $\mathcal{T} = \{\mathcal{T}_1, \ldots \mathcal{T}_g\}$ of the set of users $\mathcal{U}$, such that each $\mathcal{T}_\ell$ consists of at least t users. Then we can construct a covering design with t-anonymity sets.*

Proof. We can construct a covering design with t-anonymity sets by the following method. First, we construct a covering design on a set of g points, say $\mathcal{X} = \{x_1, \ldots, x_g\}$. We then define a bijection σ between the set of g points and the g anonymity sets, so $\sigma(\mathcal{X}) = \mathcal{T}$. Finally, for each $x_\ell \in \mathcal{X}$, we replace the point x_ℓ by the anonymity set $\sigma(x_\ell) = \mathcal{T}_{\ell'}$, where $1 \leq \ell' \leq g$. This yields a covering design satisfying the desired property. □

Theorem 3.16. *Fix a covering design with permanent anonymity sets of minimum size t. Then we achieve $(\rho, c, t-c-\rho)$-anonymity in Protocol 5 and $(\rho, c, t-c)$-anonymity in Protocol 6:*

Proof. Let C be a coalition of size at most c and consider a set of linked queries $q_1, \ldots, q_\rho$. Let S_{h_ℓ} be the memory space used for the query q_ℓ and let U_{h_ℓ} denote the proxy for q_ℓ, for $1 \leq \ell \leq \rho$.

In Protocol 5, we have

$$\begin{aligned}&\left|\left(S_{h_1} \setminus (C \cup \{U_{h_1}\})\right) \cap \left(S_{h_2} \setminus (C \cup \{U_{h_2}\})\right) \cap \cdots \cap \left(S_{h_\rho} \setminus (C \cup \{U_{q_\rho}\})\right)\right| \\ &\quad = |(S_{h_1} \cap S_{h_2} \cap \cdots \cap S_{h_\rho}) \setminus (C \cup \{U_{h_1}, U_{h_2}, \ldots, U_{h_\rho}\})| \geq t - c - \rho.\end{aligned}$$

In Protocol 6, we have

$$|(S_{h_1} \setminus C) \cap (S_{h_2} \setminus C) \cap \cdots \cap (S_{h_\rho} \setminus C)| = |(S_{h_1} \cap S_{h_2} \cap \cdots \cap S_{h_\rho}) \setminus C| \geq t - c.$$

This completes the proof. □

The idea of using permanent anonymity sets changes the trust requirements of the scheme. In particular, U_i must trust the users contained in $\mathcal{T}_i$ to a greater extent than users in $\mathcal{U} \setminus \mathcal{T}_i$, since members of $\mathcal{T}_i$ necessarily have access to U_i's query sphere. That is, there is no confidentiality among members of an anonymity set.

3.6.2.2. *Query hops.* Another possible method to increase privacy against other users, which we briefly discuss here, involves the introduction of *query hops* into the protocols. That is, we can consider allowing a designated proxy to rewrite a given query to another memory space, rather than simply forwarding the query to DB. We can establish a probabilistic approach, such that a designated proxy U_j forwards the query to DB with some fixed probability p; otherwise U_j rewrites the query uniformly at random to one of his associated memory spaces. When a response is received, a user simply posts the response back to the memory space where it was read from. This can continue until the query response reaches the source. In this case, it is easy to see that, on average, a query is posted $1/p$ times. This method removes the certainty a curious user has that the source of a given query is associated with the memory space in which that query is written. It is an interesting problem to analyze the privacy guarantees such a scheme provides against other users.

Here we consider an implementation of the protocol using a BIBD for the underlying P2P network. The protocol and analysis can be generalized to other types of designs (in the same manner as for Protocols 3–4), but we choose a BIBD for simplicity. Formally, we have the following query submission protocol:

Protocol 7. *Multi-hop BIBD Protocol*
We fix a (v, b, r, k, λ)-BIBD and let p be a fixed probability. Users submit queries according to the following steps:

1. When a user U_i wishes to send a query q to DB, he chooses to act as his own proxy with probability $1/v$. User U_i then writes the query q uniformly at random to one of the r memory spaces with which he is associated, with himself listed as proxy. Otherwise, U_i chooses uniformly at random one of the r memory spaces with which he is associated, say S_ℓ, and then he chooses uniformly at random a user $U_j \in S_\ell \backslash \{U_i\}$. Finally, user U_i requests that user U_j act as his proxy using the memory space S_ℓ.
2. When a user U_j sees that he is a requested proxy for query q in S_ℓ, the user U_j submits q directly to DB with probability p. Otherwise U_j executes Step 1, acting in the role of the source.

Suppose we have a query q posted to memory space S_ℓ with proxy U_j. We represent this event as a tuple (q, S_ℓ, U_j). We refer to each time a designated proxy U_j executes Step 1 (and rewrites a query q) as a *hop*. Let $\mathbf{H}$ be a random variable denoting the hop count. For a given query tuple (q, S_ℓ, U_j), we let $\mathbf{H}(q, S_\ell, U_j)$ denote the number of hops the query q has already taken. That is, we set $\mathbf{H}(q, S_\ell, U_j) = 0$ when the query q is first submitted by the source U_i and we increment the hop count by one every time a designated proxy rewrites the query. In particular, we are interested in computing the probability that a given query tuple (q, S_ℓ, U_j) satisfies $\mathbf{H}(q, S_\ell, U_j) = 0$, i.e., the probability that this is the first time the query q has been written to a memory space. We sometimes abuse notation and write $\mathbf{H}(q)$ for $\mathbf{H}(q, S_\ell, U_j)$ when the particular memory space S_ℓ and proxy U_j used is not relevant.

We show the probabilistic advantage a user U_t has in guessing the source of a given query in the following theorem:

Theorem 3.17. *Fix a (v, k, λ)-BIBD in Protocol 7 and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Suppose $U_t \in S_\ell$ sees a query q posted to S_ℓ with proxy $U_j \in S_\ell$ that is not his own. Then, from the point of view of U_t, for a given query q and $U_i \in \mathcal{U}\backslash\{U_t\}$, it holds that*

$$\Pr[\mathbf{S} = U_i \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j)] = \begin{cases} \frac{1-p}{v-1} + \frac{p}{k-1} & \text{if } U_i \in S_\ell \text{ and } t = j \\ \frac{1-p}{v-1} + \frac{p\lambda}{\lambda + r(k-2)} & \text{if } U_i \in S_\ell \text{ and } i = j \\ \frac{1-p}{v-1} + \frac{pr}{\lambda + r(k-2)} & \text{if } U_i \in S_\ell \text{ and } i, t \neq j \\ \frac{1-p}{v-1} & \text{if } U_i \in \mathcal{U}\backslash S_\ell \end{cases}.$$

Proof. For a population of N queries $q_1, \ldots, q_N$ with $\mathbf{H} = 0$, the expected number of queries with $\mathbf{H} = \ell$ is $N(1-p)^\ell$. That is, the expected number of queries that are generated from $q_1, \ldots, q_N$ is $N \sum_{\ell=0}^{\infty} (1-p)^\ell = N/p$. This yields

$$\Pr[\mathbf{H} = 0 \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j)] = \frac{N}{\left(\frac{N}{p}\right)} = p.$$

Consider a user $U_t \in S_\ell$ who observes the query tuple (q, S_ℓ, U_j). Then by Theorem 3.5 we have, for $U_i \neq U_t$ such that $U_i \in S_\ell$,

$$\Pr[\mathbf{S} = U_i \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j), \mathbf{H} = 0] = \begin{cases} \frac{1}{k-1} & \text{if } t = j \\ \frac{\lambda}{\lambda + r(k-2)} & \text{if } i = j \\ \frac{r}{\lambda + r(k-2)} & \text{if } i, t \neq j \end{cases}.$$

In addition, it is clear from the protocol description that a query tuple (q, S_ℓ, U_j) with nonzero hop count provides no additional information to U_t as to the identity of the source of q; that is, for all $U_i \in \mathcal{U} \backslash \{U_t\}$, we have

$$\Pr[\mathbf{S} = U_i \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j), \mathbf{H} \neq 0] = \frac{1}{v-1}.$$

We can compute the conditional distribution on possible sources $U_i \neq U_t$ by taking the weighted average of the above two probability distributions. That is,

$$\Pr[\mathbf{S} = U_i \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j)] = \begin{cases} \frac{1-p}{v-1} + \frac{p}{k-1} & \text{if } U_i \in S_\ell \text{ and } t = j \\ \frac{1-p}{v-1} + \frac{p\lambda}{\lambda + r(k-2)} & \text{if } U_i \in S_\ell \text{ and } i = j \\ \frac{1-p}{v-1} + \frac{pr}{\lambda + r(k-2)} & \text{if } U_i \in S_\ell \text{ and } i, t \neq j \\ \frac{1-p}{v-1} & \text{if } U_i \in \mathcal{U} \backslash S_\ell \end{cases}$$

This completes the proof. □

Remark 3.13. We can generalize Theorem 3.17 to coalitions C of c users by using the results of Theorem 3.6, if desired.

Theorem 3.17 indicates that as the probability p of forwarding queries directly to the database approaches zero, the probability distribution on the $v-1$ possible sources other than U_t approaches the uniform distribution. That is, the greater the likelihood that queries will be written to multiple memory spaces before being forwarded to the database, the less the advantage of U_t in identifying the source, and correspondingly, the harder the attack on linked queries described in Section 3.6.1 is to perform. Of course, we cannot completely avoid this attack on linked queries, because as before, we expect the source of a series of linked queries to appear less frequently as proxy than every other user. We make this explicit by generalizing Theorem 3.7 to the multi-hop BIBD protocol setting in the next result.

Theorem 3.18. *Fix a (v, k, λ)-BIBD in Protocol 7 and assume that $\Pr[\mathbf{S} = U_i] = 1/v$ for all $U_i \in \mathcal{U}$. Consider a user U_i and a memory space S_ℓ satisfying $U_i \in S_\ell$. Let $q_1, \ldots, q_\rho$ be a series of queries with source U_i posted to S_ℓ. The user U_i acts as his own proxy an average of $\rho\left(\frac{p}{v} + \frac{1-p}{k}\right)$ times, and every other user $U_j \neq U_i \in S_\ell$ acts as proxy an average of $\rho\left(\frac{pr}{\lambda v} + \frac{1-p}{k}\right)$ times.*

Proof. We assume $U_j \in S_\ell$. Consider a single query q. For Protocol 7, we remark that when $\mathbf{H} = 0$, we have the same analysis as in Theorem 3.7. In particular, we have

$$\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{M} = S_\ell, \mathbf{H} = 0] = \frac{\Pr[\mathbf{P} = U_j, \mathbf{S} = U_i, \mathbf{M} = S_\ell \mid \mathbf{H} = 0]}{\Pr[\mathbf{S} = U_i, \mathbf{M} = S_\ell \mid \mathbf{H} = 0]}$$
$$= \begin{cases} \frac{1}{v} & \text{if } i = j \\ \frac{r}{\lambda v} & \text{if } i \neq j \end{cases}.$$

When $\mathbf{H} \neq 0$, however, the proxy distribution is uniform, so we have

$$\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{M} = S_\ell, \mathbf{H} \neq 0] = \frac{1}{k}.$$

Recall, as shown in the proof of Theorem 3.17, $\Pr[\mathbf{H} = 0 \mid (\mathbf{Q}, \mathbf{M}, \mathbf{P}) = (q, S_\ell, U_j)] = p$. Therefore, we have

$$\Pr[\mathbf{P} = U_j \mid \mathbf{S} = U_i, \mathbf{M} = S_\ell] = \begin{cases} \frac{p}{v} + \frac{1-p}{k} & \text{if } i = j \\ \frac{pr}{\lambda v} + \frac{1-p}{k} & \text{if } i \neq j \end{cases}.$$

This completes the proof. □

3.7. Discussion and Comparison with Related Work

The concept of P2P UPIR was introduced by Domingo-Ferrer and Bras-Amorós [27] and later extended by Domingo-Ferrer et al. [28]. Stokes and Bras-Amorós [71, 73] note weaknesses in the original protocol and propose their own version. We give both the original DBWM Protocol (Protocol 1) and modified version by Stokes and Bras-Amorós (Protocol 2) in Section 3.4, and discuss the differences between these protocols and our own in detail. Furthermore, we give a detailed model for P2P UPIR in Section 3.3 and point out relevant differences with respect to related work; we reiterate that a major difference in our security model is that we assume users can communicate outside of the P2P UPIR protocol.

As already observed, previous work focuses on using configurations for the underlying P2P network, and Stokes and Bras-Amorós [71, 73] argue that the finite projective planes are the optimal choice. The relevant mathematical problems and analysis with respect to combinatorial configurations are given in detail by Bras-Amorós et al. [5]. Subsequent to our work [77], Stokes and Bras-Amorós [72] survey how to choose configurations for the DBWMS Protocol (Protocol 2). Much of this material [5, 71–73] also appears in Stokes' Ph.D. dissertation [70].

In particular, Stokes and Bras-Amorós discuss configurations which have *n-anonymous neighborhoods*, based on the well-known concept of k-anonymity [78] and similar to our notion of t-anonymity sets. The authors consider n-anonymity for $n < v$ with respect to the database (as opposed to perfect anonymity, or v-anonymity, which we consider) and propose using transversal designs with $\lambda = 1$ (defined in Section 1.2.4) to achieve

n-anonymity. Stokes and Farrás [74] build on our work [77] and that of Stokes and Bras-Amorós [72] and once again present combinatorial configurations (specifically, BIBDs and transversal designs with $\lambda = 1$) as the optimal choice for the underlying P2P network. We remark that both of these works [72, 74] assume that users can communicate only *within* shared memory spaces, which is the motivation behind limiting the P2P network topology to designs with $\lambda = 1$. The protocols provided by these authors are vulnerable to the same intersection attacks by coalitions of users we describe in Section 3.6, provided we allow for communication between users outside the confines of the chosen configuration.

Other methods of obfuscating query profiles using user collaboration have been proposed; we briefly discuss the most relevant here. Reiter and Rubin [56] introduced Crowds, which is similar in flavor to the multi-hop BIBD protocol (Protocol 7). In Crowds, a group of users send their web server requests to some other user with a fixed probability p_f and submit the request to the end server themselves with probability $1 - p_f$. Similarly, users who receive a request forward this request to another user with probability p_f and to the end server otherwise. Responses are sent back along the same path. Crowds differs from the P2P UPIR model in that there are no memory spaces, and consequently users know who has forwarded them a particular query. The *predecessor attack* on Crowds, which is identified by Reiter and Rubin [56] and discussed in detail by Wright et al. [82, 83], is a weakness of the protocol against coalitions of other users. The attack is based on the observation that the request initiator is more likely to appear directly before the first attacker on the path than any other user. Therefore, given sufficiently many rounds, a coalition of users can successfully identity the initiator of a *recurring connection* with high probability, where a recurring connection is some repeated request to an end server that can be uniquely identified by the attackers. This is similar to the attacks on linked queries in our P2P UPIR protocols we discuss in Section 3.6.

Castellà-Roca et al. [7] present a protocol that uses a similar model to that of UPIR, but only superficially analyzes the level of privacy achieved. This protocol uses a central node and cryptographic primitives in order for groups of n users to submit a set of n queries on behalf of each other; groups can only be formed once sufficiently many memberships requests have been received and are dissolved after each round of queries. Viejo and Castellà-Roca [80] present a protocol in which users exchange queries with their friends on an existing social network which has *private relationships*, i.e., the network topology is unknown. The protocol has built-in incentives to ensure good behavior among users, and the end result is that the query profile of a given user is dispersed among his neighbors. Other work [25, 42, 60, 79] explores privacy-preserving mechanisms involving submission of random queries to distort user query profiles. GoogleSharing [37] re-routes user requests to Google through proxies.

Domingo-Ferrer [26] has developed the concept of *coprivacy*, which uses game-theoretic notions to analyze user cooperation in maintaining privacy. Specifically, Domingo-Ferrer considers protocols which are *coprivate* in that an individual user's best option to maintain

his own privacy is to help other users' preserve their privacy. The author applies the theory of co-privacy to the setting of both P2P UPIR and online social networking. Recently, Domingo-Ferrer and González-Nicolás [29] extend this game-theoretic approach to analyzing P2P UPIR protocols (termed *P2P profile obfuscation protocols* in this work). The authors establish an entropy-based metric for user privacy and give conditions under which it is rational for individual users to help other users maintain privacy. Along similar lines, Rebollo-Monedero et al. [55] consider query profile obfuscation through query exchange between two users, carefully modelling user privacy in terms of the Shannon entropy of the users' apparent query profiles and defining optimization strategies for determining which queries should be exchanged.

3.8. Concluding Remarks and Future Work

In this chapter, we have given an overview and analysis of current research in UPIR, including introducing an attack by the database on user privacy. We have established a new model for P2P UPIR and considered the problem of user privacy against other users in detail, going well beyond previous work. We have given two new P2P UPIR protocols (and generalizations of these) and provided an analysis of the privacy properties provided by these protocols. In particular, our protocols have the nice property that security against the database is unconditional, in the sense that from the database's perspective, each user is equally likely to be the source of a given query (or set of linked queries). In addition, our protocols take advantage of the wide variety of available combinatorial designs. Doing so provides flexibility in the set-up phase, allowing for a choice between having a dynamic scheme (in which users are permitted to enter and leave the system), or providing increased privacy against other users.

Future work includes developing realistic cryptographic primitives for achieving the security assumptions made in our P2P UPIR protocols, such as the need for users to be unaware of which other user posted a given query to a shared memory space, and incorporating traffic analysis into the adversarial model. Moreover, developing more effective methods to mitigate attacks by user coalitions on linked queries is desirable. To this end, a more in-depth analysis of the relationship our protocols have with other profile obfuscation schemes in the literature might be informative, in particular with respect to the implications our mathematical analysis has on the efficiency of attacks and suggested preventative measures.

Another interesting avenue is to consider using transversal designs to construct the underlying P2P network, an idea contemplated by Stokes and Farrás [74] for $\lambda = 1$. In particular, using transversal designs (not necessarily with $\lambda = 1$) might be a good way to achieve scalability of the system to handle large groups of users, while still retaining acceptable levels of anonymity with respect to the database.

CHAPTER 4

Combinatorial Solutions Providing Improved Security for the Generalized Russian Cards Problem

4.1. Introduction

Suppose X is a deck of n cards, and we have three participants, Alice, Bob and Cathy. Let $a+b+c=n$ and suppose that Alice is dealt a *hand* of a cards, Bob is dealt a hand of b cards and Cathy is dealt a hand of c cards. These hands are random and dealt by some entity external to the scheme. We denote Alice's hand by H_A, Bob's hand by H_B and Cathy's hand by H_C. Of course it must be the case that $H_A \cup H_B \cup H_C = X$. We refer to this as an (a,b,c)-*deal* of the cards.

As before, for a positive integer t, let $\binom{X}{t}$ denote the set of $\binom{n}{t}$ t-subsets of X. An *announcement* by Alice $\mathcal{A}$ is a subset of $\binom{X}{a}$. It is required that when Alice makes an announcement $\mathcal{A}$, the hand she holds is one of the a-subsets in $\mathcal{A}$. The goal of the scheme is that, after a deal has taken place and Alice has made an announcement, Bob should be able to determine Alice's hand, but Cathy should not be able to determine if Alice holds any particular card not held by Cathy. These notions will be formalized as we proceed. We focus on the scenario of Bob learning Alice's hand, although the original version of this problem is for Bob and Alice to learn each other's hand. We omit the latter case, since for any protocol whereby Bob may learn Alice's hand, Bob may then announce Cathy's hand publicly. This second step provides sufficient information for Alice to determine Bob's hand, without giving Cathy any more information than she previously had.

This problem was first introduced in the case $(a,b,c)=(3,3,1)$ in the 2000 Moscow Mathematics Olympiad. Since then, there have been numerous papers investigating the problem (called the Russian cards problem) and generalizations of it, which we discuss in some detail in Section 4.8. Some are interested in card deal protocols that allow players to agree on a common secret without a given eavesdropper being able to determine this secret value. This area of research is especially interesting in terms of possible applications to key generation [**3**, **31**–**35**, **47**, **51**]. Others are concerned with analyzing variations of the problem using epistemic logic [**15**, **19**–**21**]. Duan and Yang [**30**] and He and Duan [**41**] consider a special generalization, with $n-1$ players each dealt n cards, and one player

Much of the material in this chapter appears in the paper "Combinatorial solutions providing improved security for the generalized Russian cards problem" [**75**], published in Designs, Codes and Cryptography (2012).

(the intruder) dealt one card; the authors give an algorithm by which a dealer, acting as a trusted third party, can construct announcements for each player. There have been some papers that take a combinatorial approach [1, 3, 4, 12]. In addition, there has been recent work [13, 22] in which protocols consisting of more than once announcement by Alice and Bob are considered, which is a generalization of the problem which we consider here, and one paper [14] that builds on our work.

We take a combinatorial point of view motivated by cryptographic considerations. To be specific, we provide definitions based on security conditions in the unconditionally secure framework, phrased in terms of probability distributions regarding information available to the various players (analogous to Shannon's definitions relating to perfect secrecy of a cryptosystem). In particular, we provide a formal mathematical presentation of the generalized Russian cards problem. We introduce rigorous mathematical definitions of security, which in turn allow for systematic and thorough analysis of proposed protocols. We give necessary conditions and provide constructions for schemes that satisfy the relevant definitions. Here there is a natural interplay with combinatorics, particularly the field of combinatorial designs.

4.2. Overview of Contributions

The main contributions of our work are as follows:

- We provide a formal mathematical presentation of the generalized Russian Cards problem. In particular, we define an *announcement strategy* for Alice, which designates a probability distribution on a fixed set of possible announcements Alice can make, say $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$. In keeping with standard practice in cryptography (i.e., Kerckhoffs' principle), we assume that Alice's announcement strategy is public knowledge. Security through obscurity is not considered an effective security method, as secrets are difficult to keep; providing security under the assumption the adversary has full knowledge of the set-up of the given scheme is therefore the goal. This allows us to define the *communication complexity* of the protocol to be $\lceil \log_2 m \rceil$ bits, since Alice need only broadcast the index i of her chosen announcement, which is an integer between 1 and m. In order to minimize the communication complexity of the scheme, our goal will be to minimize m, the number of possible announcements.
- We distinguish between *deterministic* strategies, in which the hand H_A held by Alice uniquely determines the index i that she will broadcast, and *non-deterministic*, possibly even biased announcement strategies. We are especially interested in strategies with uniform probability distributions, which we refer to as *equitable* strategies.
- We examine necessary and sufficient conditions for a strategy to be *informative for Bob* (i.e., strategies that allow Bob to determine Alice's hand). In particular, we give a lower bound on the number of announcements m for informative strategies and provide a nice combinatorial characterization of strategies that meet this bound, which we term *optimal* strategies.

- We provide the first formal security definitions that account for both *weak* and *perfect* security in an unconditionally secure framework. Current literature focuses on weak security. Here weak and perfect security are defined with respect to individual cards. If a scheme satisfies weak security (which we term weak 1-security), Cathy should not be able to say whether a given card is held by Alice or Bob; if a scheme satisfies perfect security (which we term perfect 1-security), each card is equally likely to be held by Alice. When Alice's strategy is equitable, we show an equivalence between perfectly secure strategies and sets of 2-designs on n points with block size a.
- We use constructions and results from the field of combinatorial designs to explore equitable strategies that are simultaneously informative and perfectly secure. In particular, we analyze the case $c = a-2$ in detail, and show that strategies for $(a, b, a-2)$-deals that are simultaneously informative and perfectly secure must satisfy $c = 1$. We also show a precise characterization between *Steiner triple systems* and informative, perfectly secure $(3, n-4, 1)$-deals.
- We generalize our notions of weak and perfect security, which focus on the probability that individual cards are held by Alice, and consider instead the probability that a given set of δ cards is held by Alice; we refer to these notions as weak or perfect δ-security. We consider equitable strategies and show an equivalence between perfectly δ-secure strategies and $(c+\delta)$-designs on n points with block size a. For equitable, informative, perfectly $(a-c-1)$-secure strategies, we achieve parallel results to the $a-c=2$ case, showing $c=1$ and demonstrating an equivalence between these strategies and *Steiner systems* $S(a-1, a, n)$.
- We show how to use a t-$(n, a, 1)$-design to construct equitable (a, b, c)-strategies that are informative for Bob and perfectly $(t-c)$-secure against Cathy for any choice of c satisfying $a-c \geq t$. In particular, this indicates that if an appropriate t-design exists, it is possible to achieve perfect security for deals where Cathy holds more than one card. We present an example construction, based on *inversive planes*, for $(q+1, q^2-q-2, 2)$-strategies which are perfectly 1-secure against Cathy and informative for Bob, where q is a prime power. This is the first strategy presented in the literature that is informative for Bob and achieves perfect 1-security against Cathy for $c > 1$.
- We discuss a variation on the generalized Russian cards problem, where the card deck is first split into a piles, and Alice and Cathy's hands consist of at most one card from each pile, with Bob receiving the remaining cards. This variant admits a nice solution using *transversal designs* with $\lambda = 1$ that achieves weak $(a-2c)$-security. In particular, this solution is easy to construct and is optimal with respect to both the number of announcements and level of security achieved.

4.2.1. Chapter outline. We define the basic framework for the generalized Russian cards problem, establish the relevant notation, and provide example solutions in Section 4.3. In Section 4.4, we study and define the notion of an informative strategy. We then move to a formal discussion of secure strategies in Section 4.5, defining and studying the security of individual cards in Section 4.5.1 and the security of multiple cards in Section 4.5.2. In

Section 4.6, we explore strategies that are simultaneously informative and either weakly or perfectly δ-secure, discussing construction methods and examples in Section 4.6.1. In Section 4.7 we discuss a variant of the generalized Russian cards problem and present a solution using transversal designs. We discuss related work in Section 4.8. Finally, we give some concluding remarks in Section 4.9.

4.3. Preliminary Notation and Examples

Alice will choose a set of announcements, say $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$, such that every $H_A \in \binom{X}{a}$ is in at least one of the m announcements. For $H_A \in \binom{X}{a}$, define $g(H_A) = \{i : H_A \in \mathcal{A}_i\}$. Alice's *announcement strategy*, or more simply, *strategy*, consists of a probability distribution p_{H_A} on $g(H_A)$, for every $H_A \in \binom{X}{a}$. The set of announcements and probability distributions are fixed ahead of time and they are public knowledge. We use the phrase (a, b, c)-*strategy* to denote a strategy for an (a, b, c)-deal. In addition, we assume without loss of generality that $p_{H_A}(i) > 0$ for all $i \in g(H_A)$. To see this, note that if $p_{H_A}(i) = 0$ for some $H_A \in \binom{X}{a}$ and $i \in g(H_A)$, this means Alice will never choose $\mathcal{A}_i$ when she holds H_A. But since the set of announcements and probability distributions are public knowledge, Cathy also knows this, so there is no reason to have included H_A in the announcement $\mathcal{A}_i$.

When Alice is dealt a hand $H_A \in \binom{X}{a}$, she randomly chooses an index $i \in g(H_A)$ according to the probability distribution p_{H_A}. Alice broadcasts the integer i to specify her announcement $\mathcal{A}_i$. Because the set of announcements and probability distributions are fixed and public, the only information that is broadcast by Alice is the index i, which is an integer between 1 and m. Therefore we define the *communication complexity* of the protocol to be $\lceil \log_2 m \rceil$ bits. In order to minimize the communication complexity of the scheme, our goal will be to minimize m, the number of possible announcements.

If $|g(H_A)| = 1$ for every H_A, then we have a *deterministic* scheme, because the hand H_A held by Alice uniquely determines the index i that she will broadcast. That is to say, in a deterministic scheme, for any given hand, there is only one possible announcement that is permitted by the given strategy.

More generally, suppose there exists a constant γ such that $|g(H_A)| = \gamma$ for every H_A. Further, suppose that every probability distribution p_{H_A} is *uniform*, i.e., $p_{H_A}(i) = 1/\gamma$ for every H_A and for every $i \in g(H_A)$. We refer to such a strategy as a γ-*equitable strategy*. A deterministic scheme is just a 1-equitable strategy.

EXAMPLE 4.1. Let $X = \{0, \ldots, 6\}$. Figure 1 presents a partition of $\binom{X}{3}$ that is due to Charlie Colbourn and Alex Rosa (private communication). This yields a deterministic $(3, 3, 1)$-strategy having $m = 6$ possible announcements.

EXAMPLE 4.2. Let $X = \{0, \ldots, 6\}$. In Figure 2, we present a set of ten announcements found by Don Kreher (private communication). It can be verified that every 3-subset of X occurs in exactly two of these announcements. Therefore we have a 2-equitable $(3, 3, 1)$-strategy.

i	$\mathcal{A}_i$
1	$\{0,1,3\},\{1,2,4\},\{2,3,5\},\{3,4,6\},\{0,4,5\},\{1,5,6\},\{0,2,6\}$
2	$\{0,2,3\},\{1,3,4\},\{2,4,5\},\{3,5,6\},\{0,4,6\},\{0,1,5\},\{1,2,6\}$
3	$\{0,2,4\},\{0,3,5\},\{1,2,3\},\{0,1,6\},\{1,4,5\},\{2,5,6\}$
4	$\{0,1,2\},\{2,3,4\},\{4,5,6\},\{1,3,5\},\{0,3,6\}$
5	$\{1,2,5\},\{0,5,6\},\{1,4,6\},\{0,3,4\},\{2,3,6\}$
6	$\{3,4,5\},\{0,1,4\},\{0,2,5\},\{2,4,6\},\{1,3,6\}$

FIGURE 1. A deterministic $(3,3,1)$-strategy having a set of six possible announcements

i	$\mathcal{A}_i$
1	$\{2,5,6\},\{2,3,4\},\{1,4,5\},\{1,3,6\},\{0,4,6\},\{0,3,5\},\{0,1,2\}$
2	$\{2,5,6\},\{2,3,4\},\{1,4,6\},\{1,3,5\},\{0,4,5\},\{0,3,6\},\{0,1,2\}$
3	$\{3,4,5\},\{2,4,6\},\{1,3,6\},\{1,2,5\},\{0,5,6\},\{0,2,3\},\{0,1,4\}$
4	$\{3,4,5\},\{2,4,6\},\{1,5,6\},\{1,2,3\},\{0,3,6\},\{0,2,5\},\{0,1,4\}$
5	$\{3,4,6\},\{2,3,5\},\{1,4,5\},\{1,2,6\},\{0,5,6\},\{0,2,4\},\{0,1,3\}$
6	$\{3,4,6\},\{2,3,5\},\{1,5,6\},\{1,2,4\},\{0,4,5\},\{0,2,6\},\{0,1,3\}$
7	$\{3,5,6\},\{2,4,5\},\{1,3,4\},\{1,2,6\},\{0,4,6\},\{0,2,3\},\{0,1,5\}$
8	$\{3,5,6\},\{2,4,5\},\{1,4,6\},\{1,2,3\},\{0,3,4\},\{0,2,6\},\{0,1,5\}$
9	$\{4,5,6\},\{2,3,6\},\{1,3,4\},\{1,2,5\},\{0,3,5\},\{0,2,4\},\{0,1,6\}$
10	$\{4,5,6\},\{2,3,6\},\{1,3,5\},\{1,2,4\},\{0,3,4\},\{0,2,5\},\{0,1,6\}$

FIGURE 2. An equitable $(3,3,1)$-strategy having a set of ten possible announcements

The following notation will be useful in formally defining *informative* and *secure* strategies $\mathcal{A}$. For any subset $Y \subseteq X$ and any announcement $\mathcal{A}_i \in \mathcal{A}$, define

$$\mathcal{P}_{\mathcal{A}}(Y,i) = \{H_A \in \mathcal{A}_i : H_A \cap Y = \emptyset\}.$$

That is, $\mathcal{P}_{\mathcal{A}}(Y,i)$ is the set of hands of $\mathcal{A}_i$ that do not intersect the subset Y. When the strategy $\mathcal{A}$ is clear from context, we write $\mathcal{P}_{\mathcal{A}}(Y,i)$ as $\mathcal{P}(Y,i)$

4.4. Informative Strategies

In this section, we formalize the notion of strategies that are *informative for Bob*, which we first introduced in Section 4.2. We first consider an (a,b,c)-deal from Bob's point of view, after hearing Alice's announcement. Suppose that $H_B \in \binom{X}{b}$ and $i \in \{1,\ldots,m\}$. Then

$$\mathcal{P}(H_B,i) = \{H_A \in \mathcal{A}_i : H_A \cap H_B = \emptyset\}.$$

That is, $\mathcal{P}(H_B,i)$ denotes the set of *possible hands* that Alice might hold, given that Bob's hand is H_B and Alice's announcement is $\mathcal{A}_i$. Note that if Alice chooses the announcement

$\mathcal{A}_i$, then $\mathcal{P}(H_B, i) \neq \emptyset$, as Alice's strategy requires that her hand be an element of $\mathcal{A}_i$. Alice's strategy is *informative for Bob* provided that

$$|\mathcal{P}(H_B, i)| \leq 1 \tag{14}$$

for all $H_B \in \binom{X}{b}$ and for all i. In this situation, if Bob holds the cards in H_B and Alice broadcasts i, then Bob can determine the set of a cards that Alice holds.

If for a particular announcement $\mathcal{A}_i$ and any hand $H_B \in \binom{X}{b}$, we have that $|\mathcal{P}(H_B, i)| \leq 1$, we say that $\mathcal{A}_i$ is an *informative announcement*. This terminology is in keeping with previous work, which considers protocol characteristics only on the level of individual announcements.

The following result was shown by Albert et al. [1], albeit using different terminology:

Theorem 4.1. *The announcement $\mathcal{A}_i$ is informative for Bob if and only if there do not exist two distinct sets H_A, $H'_A \in \mathcal{A}_i$ such that $|H_A \cap H'_A| \geq a - c$.*

Proof. Suppose there exist two distinct sets $H_A, H'_A \in \mathcal{A}_i$ such that $|H_A \cap H'_A| \geq a - c$. We have that $|H_A \cup H'_A| \leq 2a - (a - c) = a + c = n - b$. Hence, there exists $H_B \in \binom{X}{b}$ such that $H_B \cap (H_A \cup H'_A) = \emptyset$. Then $\{H_A, H'_A\} \subseteq \mathcal{P}(H_B, i)$, which contradicts (14).

Conversely, suppose $\{H_A, H'_A\} \subseteq \mathcal{P}(H_B, i)$, where $H_A \neq H'_A$. Then $|H_A \cup H'_A| \leq n - b = a + c$, and hence $|H_A \cap H'_A| \geq a - c$. □

It follows from Theorem 4.1 that the $(3, 3, 1)$-strategies presented in Examples 4.1 and 4.2 are both informative for Bob, because $|H_A \cap H'_A| \leq 1$ whenever H_A and H'_A are two distinct sets in the same announcement.

We also have the following necessary condition.

Corollary 4.2. *Suppose there exists a strategy for Alice that is informative for Bob. Then $a > c$.*

Furthermore, when $a > c$, we can derive a lower bound on the size of Alice's announcement.

Theorem 4.3. *Suppose $a > c$ and there exists a strategy for Alice that is informative for Bob. Then $m \geq \binom{n-a+c}{c}$.*

Proof. Let $X' \subseteq X$ where $|X'| = a - c$. There are precisely $\binom{n-a+c}{c}$ a-subsets of X that contain X'. These a-subsets must occur in different announcements, by Theorem 4.1. Therefore $m \geq \binom{n-a+c}{c}$. □

In view of the above theorem, an (a, b, c)-strategy for Alice that is informative for Bob is said to be *optimal* if $m = \binom{n-a+c}{c}$. In fact, we can give a nice combinatorial characterization of such optimal strategies. First we make the following observation, which follows directly from Theorem 4.1 and the definition of a t-design.

Corollary 4.4. *Suppose $a > c$ and each announcement in an (a, b, c)-strategy is a t-$(n, a, 1)$-design, where $t \leq a - c$. Then the strategy is informative for Bob.*

We have the following combinatorial characterization of optimal strategies:

Theorem 4.5. *Suppose that $a > c$. An optimal (a, b, c)-strategy for Alice that is informative for Bob is equivalent to a large set of t-$(n, a, 1)$-designs, where $t = a - c$.*

Proof. Suppose there exists a large set of $(a - c)$-$(n, a, 1)$-designs. Recall from Definition 1.28 that the set of all blocks sets (i.e., possible announcements) in this large set form a partition of $\binom{X}{a}$ and that there are precisely $\binom{n-a+c)}{c}$ designs in such a set. Then it is easy to see that this immediately yields an optimal (a, b, c)-strategy for Alice that is informative for Bob.

Conversely, suppose there is an optimal (a, b, c)-strategy for Alice that is informative for Bob. We need to show that every announcement is an $(a - c)$-$(n, a, 1)$-design. Denote $t = a - c$ and let $X' \subseteq X$, $|X'| = t$. From the proof of Theorem 4.3, the a-subsets containing X' occur in $\binom{n-a+c}{c}$ different announcements. However, there are a total of $\binom{n-a+c}{c}$ announcements, so every announcement must contain a block that contains X'. □

An optimal $(3, 3, 1)$-strategy would have $m = 5$. From Theorem 4.5, the existence of such a strategy would be equivalent to a large set of five STS(7), or Steiner triple systems of order 7. As mentioned in Remark 1.10, it is known that this large set does not exist. However, from Example 4.1, we obtain a $(3, 3, 1)$-strategy for Alice with $m = 6$ that is informative for Bob. Thus we have proven the following.

Theorem 4.6. *The minimum m such that there exists a $(3, 3, 1)$-strategy for Alice that is informative for Bob is $m = 6$.*

It is possible to have informative (a, b, c)-strategies using announcements which are t-designs with $\lambda > 1$. In particular, Theorem 4.1 indicates that the block intersection properties of the chosen design are relevant to whether or not the strategy is informative. If every announcement is a *symmetric BIBD*, for example, then the strategy is guaranteed to be informative when $a - c > \lambda$. This is because the intersection of any two blocks in a symmetric BIBD contains exactly λ points, as stated in Theorem 1.4.

We make one more observation relating combinatorial designs and informative strategies.

Lemma 4.7. *Suppose $a > c$ and each announcement $\mathcal{A}_i$ in an (a, b, c)-strategy $\mathcal{A}$ is a t_i-(n, a, λ_i)-design, where $t_i \geq a - c$. If $\mathcal{A}$ is informative for Bob, then $t_i = a - c$ and $\lambda_i = 1$ for all i.*

Proof. Consider an announcement $\mathcal{A}_i \in \mathcal{A}$. If $\lambda_i > 1$, then there exist two blocks whose intersection has cardinality at least $t_i \geq a - c$. This contradicts Theorem 4.1, so $\lambda_i = 1$, as desired.

If $t_i > a - c$, then from Theorem 1.6, there are

$$\frac{v - (t_i - 1)}{k - (t_i - 1)} > 1$$

blocks that contain $t_i - 1$ fixed points. Since $t_i - 1 \geq a - c$, this contradicts Theorem 4.1, so $t_i = a - c$, as desired. □

4.5. Secure Strategies

In this section, we formalize the notion of strategies that are *secure against Cathy*, which we first introduced in Section 4.2. Suppose that Alice makes an announcement $\mathcal{A}_i$ while trying to conceal information about her hand from Cathy. Necessarily Alice's hand is an a-subset in $\mathcal{A}_i$. In fact, Cathy knows that Alice's hand must be one of the a-subsets in the set $\mathcal{P}(H_C, i) = \{H_A \in \mathcal{A}_i : H_A \cap H_C = \emptyset\}$. Therefore Cathy does obtain some partial information about Alice's hand. However, it might be possible to prevent Cathy from determining whether any individual card (or perhaps some subset of cards) in $X \backslash H_C$ is held by Alice or by Bob. For readability, we first treat the security of individual cards in Section 4.5.1, before generalizing to the security of multiple cards in Section 4.5.2.

4.5.1. Security of individual cards. We define two notions for the security of individual cards with respect to Cathy:

Definition 4.1.

1. Alice's strategy is *weakly* 1*-secure against Cathy* provided that, for any announcement $\mathcal{A}_i$, for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any $x \in X \backslash H_C$, it holds that
$$0 < \Pr[x \in H_A \mid i, H_C] < 1.$$
Weak security means that, from Cathy's point of view, any individual card in $X \backslash H_C$ could be held by either Alice or Bob.
2. Alice's strategy is *perfectly* 1*-secure against Cathy* provided that for any announcement $\mathcal{A}_i$, for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any $x \in X \backslash H_C$, it holds that
$$\Pr[x \in H_A \mid i, H_C] = \frac{a}{a + b}.$$
Perfect security means that, from Cathy's point of view, the probability that any individual card in $X \backslash H_C$ is held by Alice is a constant. This probability must equal $a/(a + b)$ because Alice holds a of the $a + b$ cards not held by Cathy.

It is obvious that perfect 1-security implies weak 1-security.

REMARK 4.1. The condition $\mathcal{P}(H_C, i) \neq \emptyset$ is included to account for the possibility that an announcement $\mathcal{A}_i$ is not compatible (i.e., will never be announced) with certain hands H_C held by Cathy. That is, we wish to ensure that the conditional probability $\Pr[x \in H_A \mid i, H_C]$ is defined.

We have the following elementary result:

Lemma 4.8. *Consider an (a, b, c)-strategy $\mathcal{A}$ that is weakly 1-secure. Then for all $\mathcal{A}_i \in \mathcal{A}$ and $x \in X$, we have $\mathcal{P}(\{x\}, i) \neq \emptyset$.*

Proof. We proceed by contradiction. Suppose $\mathcal{P}(\{x\}, i) = \emptyset$ for some $\mathcal{A}_i \in \mathcal{A}$ and $x \in X$. Then x occurs in every hand of $\mathcal{A}_i$. That is, if Alice announces $\mathcal{A}_i$, then Alice must hold x. In particular, this implies that Cathy's hand, say H_C, does not contain x and $\Pr[x \in H_A \mid i, H_C] = 1$. □

The conditions for weak and perfect 1-security depend on the probability distributions p_{H_A} and the possible announcements. We derive simpler, but equivalent, conditions of a combinatorial nature when Alice's strategy is equitable. First we state and prove a useful lemma which establishes that in an equitable strategy, from Cathy's point of view, any hand $H_A \in \mathcal{P}(H_C, i)$ is equally likely.

Lemma 4.9. *Suppose that Alice's strategy is γ-equitable, Alice's announcement is $\mathcal{A}_i$, $H_C \in \binom{X}{c}$ and $H_A \in \mathcal{P}(H_C, i)$. Then*

$$\Pr[H_A \mid H_C, i] = \frac{1}{|\mathcal{P}(H_C, i)|}. \tag{15}$$

Proof. We have

$$\Pr[H_A \mid H_C, i] = \frac{\Pr[H_A, H_C, i]}{\Pr[H_C, i]}.$$

We can compute

$$\begin{aligned}\Pr[H_A, H_C, i] &= \Pr[H_C \mid H_A, i] \Pr[i \mid H_A] \Pr[H_A] \\ &= \frac{1}{\binom{b+c}{c}} \times \frac{1}{\gamma} \times \frac{1}{\binom{n}{a}}.\end{aligned}$$

Similarly, we have

$$\begin{aligned}\Pr[H_C, i] &= \sum_{H'_A \in \mathcal{P}(H_C, i)} \Pr[H_C \mid H'_A, i] \Pr[i \mid H'_A] \Pr[H'_A] \\ &= |\mathcal{P}(H_C, i)| \times \frac{1}{\binom{b+c}{c}} \times \frac{1}{\gamma} \times \frac{1}{\binom{n}{a}}.\end{aligned}$$

The result follows. □

Theorem 4.10. *Suppose that Alice's strategy is γ-equitable. Then the following hold:*

1. *Alice's strategy is weakly* 1*-secure against Cathy if and only if, for any announcement* $\mathcal{A}_i$*, for any* $H_C \in \binom{X}{c}$ *such that* $\mathcal{P}(H_C, i) \neq \emptyset$*, and for any* $x \in X \backslash H_C$*, it holds that*
$$1 \leq |\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}| \leq |\mathcal{P}(H_C, i)| - 1.$$
2. *Alice's strategy is perfectly* 1*-secure against Cathy if and only if, for any announcement* $\mathcal{A}_i$ *and for any* $H_C \in \binom{X}{c}$ *such that* $\mathcal{P}(H_C, i) \neq \emptyset$*, it holds that*
$$|\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}| = \frac{a\,|\mathcal{P}(H_C, i)|}{a+b}$$
for any $x \in X \backslash H_C$*.*

Proof. Since (15) holds, it immediately follows that
$$\Pr[x \in H_A \mid i, H_C] = \frac{|\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}|}{|\mathcal{P}(H_C, i)|}. \tag{16}$$
Using Equation (16), we observe that
$$0 < \frac{|\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}|}{|\mathcal{P}(H_C, i)|} < 1$$
holds if and only if
$$1 \leq |\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}| \leq |\mathcal{P}(H_C, i)| - 1.$$
This gives the first condition of the theorem.

Define $r_x = |\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}|$. Alice's strategy is perfectly 1-secure against Cathy if and only if the value $\Pr[x \in H_A \mid i, H_C]$ is independent of x. From (16), this occurs if and only if r_x is independent of x. We have that
$$\sum_{x \in X \backslash H_C} r_x = a\,|\mathcal{P}(H_C, i)|.$$
There are $a + b$ terms r_x in the above sum. These terms are all equal if and only if they all have the value $r = a\,|\mathcal{P}(H_C, i)|/(a+b)$. This proves the second condition of the theorem. □

REMARK 4.2. The above characterization of weak 1-security for equitable strategies is equivalent to axioms **CA2** and **CA3** in [1]. The characterization of perfect 1-security for equitable strategies is equivalent to axiom **CA4** in [4].

It can be verified that the $(3, 3, 1)$-strategy in Example 4.2 is perfectly 1-secure against Cathy. However, the $(3, 3, 1)$-strategy in Example 4.1 is only weakly 1-secure against Cathy.

Here is a sufficient condition for an equitable strategy to be perfectly 1-secure against Cathy.

Lemma 4.11. *Suppose that each announcement* $\mathcal{A}_i$ *in an equitable* $(a, b, 1)$*-strategy* $\mathcal{A}$ *is a* 2*-*(n, a, λ_i)*-design. Then the strategy is perfectly* 1*-secure against Cathy.*

Proof. Consider an announcement $\mathcal{A}_i \in \mathcal{A}$ and a possible hand $H_C = \{y\}$ for Cathy. There are

$$|\mathcal{P}(H_C, i)| = \lambda_i \left(\frac{n(n-1)}{a(a-1)} - \frac{n-1}{a-1} \right)$$

blocks in $\mathcal{A}_i$ that do not contain y. Each point $x \in X \backslash \{y\}$ is contained in precisely

$$|\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}| = \lambda_i \left(\frac{n-1}{a-1} - 1 \right)$$

of these blocks.

Then for any $x \in X \backslash \{y\}$, we have

$$\frac{|\mathcal{P}(H_C, i)|}{|\{H_A \in \mathcal{P}(H_C, i) : x \in H_A\}|} = \frac{n-1}{a} = \frac{a+b}{a},$$

so Condition 2 of Theorem 4.10 is satisfied. □

In fact, the condition that every announcement $\mathcal{A}_i$ be a 2-(n, a, λ_i)-design is also a necessary condition for an equitable $(a, b, 1)$-strategy to be perfectly 1-secure, as the following Theorem shows.

Theorem 4.12. *Suppose we have an equitable $(a, b, 1)$-strategy $\mathcal{A}$ that is perfectly 1-secure against Cathy. Then every announcement $\mathcal{A}_i \in \mathcal{A}$ is a 2-(n, a, λ_i)-design.*

Proof. First observe that since Cathy holds only one card, Lemma 4.8 immediately implies that any element $x \in X$ is a possible hand for Cathy. Consider an announcement $\mathcal{A}_i \in \mathcal{A}$. We proceed by showing that every pair of distinct elements x, $y \in X$ occurs in a constant number of hands of $\mathcal{A}_i$.

Let $x \in X$. Define r_x to be the number of hands of $\mathcal{A}_i$ containing x. We proceed by counting r_x in two different ways. On the one hand, we immediately have

$$r_x = |\mathcal{A}_i| - |\mathcal{P}(\{x\}, i)|. \tag{17}$$

On the other hand, we can relate r_x to $\mathcal{P}(\{y\}, i)$ for any $y \neq x \in X$ as follows. Since the strategy is perfectly 1-secure, x occurs a constant number of times in $\mathcal{P}(\{y\}, i)$, namely $\frac{a}{a+b} |\mathcal{P}(\{y\}, i)|$ times. In particular, this is the number of times x occurs in a hand of $\mathcal{A}_i$ without y. That is, letting λ_{xy} denote the number of times x occurs together with y in a hand of $\mathcal{A}_i$, we have

$$r_x = \lambda_{xy} + \frac{a}{a+b} |\mathcal{P}(\{y\}, i)|. \tag{18}$$

This gives us

$$|\mathcal{A}_i| = \lambda_{xy} + \frac{a}{a+b} |\mathcal{P}(\{y\}, i)| + |\mathcal{P}(\{x\}, i)|. \tag{19}$$

Now, following the same logic for y, we also have

$$|\mathcal{A}_i| = \lambda_{xy} + \frac{a}{a+b} |\mathcal{P}(\{x\}, i)| + |\mathcal{P}(\{y\}, i)|. \tag{20}$$

Equating Equations (19) and (20) shows that $|\mathcal{P}(\{x\}, i)|$ is independent of the choice of $x \in X$. That is, r_x is independent of x (by Equation 17), so every point of X occurs in a constant number of hands of $\mathcal{A}_i$, say r hands. Moreover, Equation 18 then gives

$$\lambda_{xy} = r - \frac{a}{a+b} |\mathcal{P}(\{y\}, i)| = r - \frac{a}{a+b} (|A_i| - r),$$

so λ_{xy} is independent of x and y. That is, every pair of points x, $y \in X$ occurs a constant number of times, which we denote by λ_i. This implies $\mathcal{A}_i$ is a 2-(n, a, λ_i)-design. □

As we will see once we have generalized our security notions to account for multiple cards, the relationship between combinatorial designs and strategies that satisfy our notions of perfect security is quite deep.

4.5.2. Security of multiple cards. We can generalize the definitions of weak and perfect 1-security to weak and perfect δ-security in the natural way.

Definition 4.2. Let $1 \leq \delta \leq a$.

1. Alice's strategy is *weakly δ-secure against Cathy* provided that for any δ' such that $1 \leq \delta' \leq \delta$, for any announcement $\mathcal{A}_i$, for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any δ' distinct elements $x_1, \ldots, x_{\delta'} \in X \backslash H_C$, it holds that
$$0 < \Pr[x_1, \ldots, x_{\delta'} \in H_A \mid i, H_C] < 1.$$
Weak security means that, from Cathy's point of view, any set of δ or fewer elements from $X \backslash H_C$ may or may not be held by Alice.
2. Alice's strategy is *perfectly δ-secure against Cathy* provided that for any δ' such that $1 \leq \delta' \leq \delta$, for any announcement $\mathcal{A}_i$, for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any δ' distinct elements $x_1, \ldots, x_{\delta'} \in X \backslash H_C$, it holds that
$$\Pr[x_1, \ldots, x_{\delta'} \in H_A \mid i, H_C] = \frac{\binom{a}{\delta'}}{\binom{a+b}{\delta'}}.$$
Perfect security means that, from Cathy's point of view, the probability that any set of δ or fewer cards from $X \backslash H_C$ is held by Alice is a constant.

It is obvious that perfect δ-security implies weak δ-security.

REMARK 4.3. The condition $\mathcal{P}(H_C, i) \neq \emptyset$ is included to account for the possibility that an announcement $\mathcal{A}_i$ is not compatible with certain hands H_C held by Cathy.

The conditions for weak and perfect δ-security depend on the probability distributions p_{H_A} and the possible announcements. As before, we will derive simpler, but equivalent, conditions of a combinatorial nature when Alice's strategy is equitable.

Theorem 4.13. *Suppose that Alice's strategy is γ-equitable. Then the following hold:*

1. *Alice's strategy is weakly δ-secure against Cathy if and only if, for any δ' such that $1 \leq \delta' \leq \delta$, for any announcement $\mathcal{A}_i$, for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any δ' distinct elements $x_1, \ldots, x_{\delta'} \in X \backslash H_C$, it holds that*
$$1 \leq |\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_{\delta'} \in H_A\}| \leq |\mathcal{P}(H_C, i)| - 1.$$
2. *Alice's strategy is perfectly δ-secure against Cathy if and only if, for any announcement $\mathcal{A}_i$ and for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, it holds that*
$$|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_\delta \in H_A\}| = \frac{\binom{a}{\delta} \ |\mathcal{P}(H_C, i)|}{\binom{a+b}{\delta}}$$
for any δ distinct elements $x_1, \ldots, x_\delta \in X \backslash H_C$.

Proof. Let $1 \leq \delta' \leq \delta$.

Since (15) (from Lemma 4.9) holds, it immediately follows that
$$\Pr[x_1, \ldots, x_{\delta'} \in H_A \mid i, H_C] = \frac{|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_{\delta'} \in H_A\}|}{|\mathcal{P}(H_C, i)|}. \tag{21}$$
Using Equation (21), we observe that
$$0 < \frac{|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_{\delta'} \in H_A\}|}{|\mathcal{P}(H_C, i)|} < 1$$
holds if and only if
$$1 \leq |\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_{\delta'} \in H_A\}| \leq |\mathcal{P}(H_C, i)| - 1.$$
This gives the first condition of the theorem.

For the second condition of the theorem, we first remark that, if the given security property holds for δ, it will automatically hold for δ' such that $1 \leq \delta' \leq \delta$. This is because the security property for δ says that every δ-subset occurs the same number of times within a certain set of blocks of size $|\mathcal{P}(H_C, i)|$. That is, we have a t-design with $t = \delta$. Now, by Corollary 1.7, every t-design is a t'-design for all $t' \leq t$. Thus it suffices to show that, for any announcement $\mathcal{A}_i$ and for any $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, and for any δ distinct elements $x_1, \ldots, x_\delta \in X \backslash H_C$, that
$$|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_\delta \in H_A\}| = \frac{\binom{a}{\delta} \ |\mathcal{P}(H_C, i)|}{\binom{a+b}{\delta}}$$
holds if and only if
$$\Pr[x_1, \ldots, x_\delta \in H_A \mid i, H_C] = \frac{\binom{a}{\delta}}{\binom{a+b}{\delta}}.$$

Define $r_{x_1,\ldots,x_\delta} = |\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_\delta \in H_A\}|$. Alice's strategy is perfectly δ-secure against Cathy if and only if the value $\Pr[x_1, \ldots, x_\delta \in H_A \mid i, H_C]$ is independent of

the δ-subset $\{x_1, \ldots, x_\delta\}$. From (21), this occurs if and only if $r_{x_1,\ldots,x_\delta}$ is independent of the δ-subset $\{x_1, \ldots, x_\delta\}$. We have that

$$\sum_{D \in \binom{X \setminus H_C}{\delta}} r_D = \binom{a}{\delta} \left| \mathcal{P}\left(H_C, i\right) \right|.$$

There are $\binom{a+b}{\delta}$ terms r_D in the above sum. These terms are all equal if and only if they all have the value $r = \binom{a}{\delta} \left| \mathcal{P}\left(H_C, i\right) \right| / \binom{a+b}{\delta}$. This completes the proof. □

Theorem 4.14. *Suppose that each announcement $\mathcal{A}_i$ in an equitable (a, b, c)-strategy $\mathcal{A}$ is a t-(n, a, λ_i)-design. Then the strategy is perfectly $(t - c)$-secure against Cathy.*

Proof. Consider an announcement $\mathcal{A}_i \in \mathcal{A}$ and a possible hand H_C for Cathy. Since $c \leq t$, Theorem 1.8 implies there are

$$\left| \mathcal{P}\left(H_C, i\right) \right| = \frac{\lambda_i \binom{n-c}{a}}{\binom{n-t}{a-t}} = \frac{\lambda_i \binom{a+b}{a}}{\binom{n-t}{a-t}}$$

blocks in $\mathcal{A}_i$ that do not contain any of the points of H_C.

Let $\delta \leq t - c$. Then Theorem 1.8 also implies that each set of δ points $x_1, \ldots, x_\delta \in X \setminus H_C$ is contained in precisely

$$\left| \{H_A \in \mathcal{P}\left(H_C, i\right) : x_1, \ldots, x_\delta \in H_A\} \right| = \frac{\lambda_i \binom{n-\delta-c}{a-\delta}}{\binom{n-t}{a-t}} = \frac{\lambda_i \binom{a+b-\delta}{a-\delta}}{\binom{n-t}{a-t}}$$

of these blocks.

Thus, for any set of δ points $x_1, \ldots, x_\delta \in X \setminus H_C$, we have

$$\frac{\left| \mathcal{P}\left(H_C, i\right) \right|}{\left| \{H_A \in \mathcal{P}\left(H_C, i\right) : x_1, \ldots, x_\delta \in H_A\} \right|} = \frac{(a+b)!(a-\delta)!}{a!(a+b-\delta)!} = \frac{\binom{a+b}{\delta}}{\binom{a}{\delta}},$$

so Condition 2 of Theorem 4.13 is satisfied. □

For deals satisfying $c = 1$, we have the following necessary condition for an equitable strategy to be perfectly δ-secure.

Theorem 4.15. *Suppose we have an equitable $(a, b, 1)$-strategy $\mathcal{A}$ that is perfectly δ-secure against Cathy. Then every announcement $\mathcal{A}_i \in \mathcal{A}$ is a $(\delta + 1)$-(n, a, λ_i)-design.*

Proof. We proceed by induction on δ. The base case ($\delta = 1$) is shown in Theorem 4.12.

Consider an announcement $\mathcal{A}_i \in \mathcal{A}$. For a subset $Y \subseteq X$, let λ_Y denote the number of hands of $\mathcal{A}_i$ that contain Y. We show $\mathcal{A}_i$ must be a $(\delta + 1)$-design as follows.

Suppose we have $Y \subseteq X$, where $|Y| = \delta + 1$. Pick an element $y \in Y$. Since $c = 1$, we have by Lemma 4.8 that $\{y\}$ is a possible hand for Cathy. Since $\mathcal{A}$ is equitable and perfectly δ-secure, we have (by Theorem 4.13)

$$|\{H_A \in \mathcal{P}(\{y\}, i) : Y \backslash \{y\} \subseteq H_A\}| = \frac{\binom{a}{\delta} \ |\mathcal{P}(\{y\}, i)|}{\binom{a+b}{\delta}}.$$

Moreover, since perfect δ-security implies perfect 1-security, $|\mathcal{P}(\{y\}, i)|$ is independent of y, as shown in the proof of Theorem 4.12. That is, the number of hands of $\mathcal{A}_i$ that contain the δ-subset $Y \backslash \{y\}$ but do not contain y is independent of the choice of Y and $y \in Y$, i.e. is some constant, say s.

Now, $\mathcal{A}$ must be perfectly $(\delta - 1)$-secure (since $\mathcal{A}$ is perfectly δ-secure), so by the inductive hypothesis, $\mathcal{A}_i$ is a δ-(n, a, λ_i')-design for some λ_i'. Therefore, the number of hands of $\mathcal{A}_i$ that contain the δ-subset $Y \backslash \{y\}$ is precisely λ_i'.

We have

$$\begin{aligned} \lambda_{Y \backslash \{y\}} &= \lambda_Y + \frac{\binom{a}{\delta} \ |\mathcal{P}(\{y\}, i)|}{\binom{a+b}{\delta}} \\ \iff \lambda_i' &= \lambda_Y + s. \end{aligned}$$

Therefore, λ_Y is some constant independent of Y, so every $(\delta + 1)$-subset occurs in a constant number of hands of $\mathcal{A}_i$, say λ_i. This implies $\mathcal{A}_i$ is a $(\delta + 1)$-(n, a, λ_i)-design, as desired. □

We are now ready to give a combinatorial characterization of general (a, b, c)-strategies that are equitable and perfectly δ-secure for some $\delta \geq 1$.

Theorem 4.16. *Suppose we have an equitable (a, b, c)-strategy $\mathcal{A}$ that is perfectly δ-secure against Cathy. Then every announcement $\mathcal{A}_i \in \mathcal{A}$ is a $(c + \delta)$-(n, a, λ_i)-design.*

Proof. We proceed by induction on c. The base case $c = 1$ is shown in Theorem 4.15. Recall that for a strategy $\mathcal{A}$, an announcement $\mathcal{A}_i \in \mathcal{A}$, and a subset $Y \subseteq X$, we make the strategy $\mathcal{A}$ explicit in the notation $\mathcal{P}(Y, i)$ by writing $\mathcal{P}_{\mathcal{A}}(Y, i)$.

Let $y \in X$ and define $X' = X \backslash \{y\}$. Define an $(a, b, c - 1)$-strategy $\mathcal{A}'$ by

$$\mathcal{A}' = \{\mathcal{A}_i' : \mathcal{A}_i' = \mathcal{P}_{\mathcal{A}}(\{y\}, i), A_i \in \mathcal{A}\}.$$

We now show $\mathcal{A}'$ is perfectly δ-secure. Suppose Cathy holds a $(c - 1)$-subset $Y \subseteq X'$ satisfying $\mathcal{P}_{\mathcal{A}'}(Y, i) \neq \emptyset$ for some $\mathcal{A}_i'$. In particular, note that if no such $\mathcal{A}_i'$ exists, then $\mathcal{A}'$ is trivially perfectly δ-secure.

Consider a δ-subset $Z \subseteq X' \backslash Y = X \backslash (Y \cup \{y\})$. We wish to count the number of hands in $\mathcal{P}_{\mathcal{A}'}(Y, i)$ that contain Z. Now, $\mathcal{P}_{\mathcal{A}'}(Y, i) = \mathcal{P}_{\mathcal{A}}(Y \cup \{y\}, i)$, so $\mathcal{P}_{\mathcal{A}}(Y \cup \{y\}, i) \neq \emptyset$ and

hence $Y \cup \{y\}$ is a possible hand for Cathy in the original strategy $\mathcal{A}$. Since $\mathcal{A}$ is perfectly δ-secure, we see that (by Theorem 4.13)

$$|\{H_A \in \mathcal{P}_{\mathcal{A}}(Y \cup \{y\}, i) : Z \subseteq H_A\}| = \frac{\binom{a}{\delta}\ |\mathcal{P}_{\mathcal{A}}(Y \cup \{y\}, i)|}{\binom{a+b}{\delta}},$$

which together with the fact that $\mathcal{P}_{\mathcal{A}'}(Y, i) = \mathcal{P}_{\mathcal{A}}(Y \cup \{y\}, i)$, immediately implies $\mathcal{A}'$ is perfectly δ-secure. Moreover, since $\mathcal{A}'$ is a perfectly δ-secure $(a, b, c-1)$-strategy, we have by the inductive hypothesis that every announcement $\mathcal{A}'_i \in \mathcal{A}'$ is a $(c-1+\delta)$-$(n-1, a, \lambda_i)$-design for some λ'_i (where λ'_i may depend on i).

That is, every $(c-1+\delta)$-subset of $X \backslash \{y\}$ occurs in λ'_i hands of $\mathcal{A}'_i = \mathcal{P}_{\mathcal{A}}(\{y\}, i)$. Since we chose y to be an arbitrary element of X, this implies $\mathcal{A}$ is a $(c-1+\delta)$-perfectly secure $(a, b+c-1, 1)$-strategy. Then the base case (Theorem 4.15) implies that every announcement $\mathcal{A}_i \in \mathcal{A}$ is a $(c+\delta)$-(n, a, λ_i)-design for some λ_i (where λ_i may depend on i), as desired. □

Theorem 4.16 immediately implies the following bound on the security parameter δ for equitable strategies:

Corollary 4.17. *Suppose we have an equitable (a, b, c)-strategy $\mathcal{A}$ that is perfectly δ-secure against Cathy. Then $\delta \leq a - c$.*

REMARK 4.4. If we have an equitable (a, b, c)-strategy $\mathcal{A}$ that is perfectly δ-secure against Cathy, where $\delta = a - c$, then each announcement $\mathcal{A}_i \in \mathcal{A}$ is an a-design. In fact, since every a-subset of X must appear a constant number of times in each $\mathcal{A}_i$, we see that each $\mathcal{A}_i$ is a *trivial a-design*. In this case, we see Alice's strategy is not informative for Bob.

Together, Theorem 4.14 and Theorem 4.16 show a direct correspondence between t-designs and equitable announcement strategies that are perfectly δ-secure for some δ satisfying $\delta \leq t - c$. We state this result in the following Theorem for clarity.

Theorem 4.18. *A γ-equitable (a, b, c)-strategy $\mathcal{A}$ on card deck X that is perfectly δ-secure against Cathy is equivalent to a set of $(c+\delta)$-designs with point set X and block size a having the property that every a-subset of X occurs in precisely γ of these designs.*

4.6. Simultaneously Informative and Secure Strategies

In general, we want to find an (a, b, c)-strategy (for Alice) that is simultaneously informative for Bob and (perfectly or weakly) δ-secure against Cathy. We first consider informative strategies that provide security for individual cards and then consider informative strategies that provide security for multiple cards.

The following was first shown by Albert et al. [1] using a different proof technique:

Theorem 4.19. *If $a \leq c+1$, then there does not exist a strategy for Alice that is simultaneously informative for Bob and weakly 1-secure against Cathy.*

Proof. In view of Corollary 4.2, we only need to consider the case $a = c + 1$. In this case, any two a-subsets in an announcement must be disjoint, by Theorem 4.1. For any announcement $\mathcal{A}_i$ and any $x \in X$, the definition of weak 1-security necessitates the existence of a block in $\mathcal{A}_i$ that contains x. It therefore follows that every $\mathcal{A}_i$ forms a partition of X into n/a blocks.

Now, suppose that Alice's announcement is $\mathcal{A}_i$ and Cathy's hand is H_C. There exists at least one $H_A \in \mathcal{A}_i$ such that $H_A \cap H_C \neq \emptyset$. Now, $|H_C| < |H_A|$, so there is a point $x \in H_A \backslash H_C$. The existence of this point violates the requirement of weak 1-security. □

It is worth observing that a strategy that is not informative for Cathy implies, for any announcement $\mathcal{A}_i$ by Alice and possible hand $H_C \in \binom{X}{c}$ such that $\mathcal{P}(H_C, i) \neq \emptyset$, that $|\mathcal{P}(H_C, i)| \geq 2$. That is, there must exist distinct $H_A, H'_A \in \mathcal{P}(H_C, i)$. Following the same technique as in the proof of Lemma 4.1, this implies $|H_A \cap H'_A| \geq a - b$. If in addition the strategy is informative for Bob, by Lemma 4.1 we have $a - c > |H_A \cap H'_A| \geq a - b$, so $c < b$. This gives us the following result (which is also discussed by Albert et al. [1]):

Theorem 4.20. *If $c \geq b$, then there does not exist a strategy for Alice that is simultaneously informative for Bob and weakly 1-secure against Cathy.*

We now focus on $(3, n-4, 1)$-deals and examine the relationship between informative and perfectly 1-secure strategies and Steiner triple systems. We begin with some existence results for optimal strategies.

Theorem 4.21. *Suppose $(a, b, c) = (3, n-4, 1)$, where $n \equiv 1, 3 \bmod 6$, $n > 7$. Then there exists an optimal strategy for Alice that is informative for Bob and perfectly 1-secure against Cathy.*

Proof. If $n \equiv 1, 3 \bmod 6$, $n > 7$, then there exists a large set of disjoint STS(n) on an n-set X (as mentioned in Remark 1.10). Theorem 4.5 establishes that the resulting strategy is informative for Bob, because no announcement $\mathcal{A}_i$ (the set of blocks of an STS(n)) contains two blocks that intersect in more than one point. Perfect 1-security follows immediately from Lemma 4.11. Note that this strategy is optimal, since every a-subset occurs in exactly one announcement. □

Example 4.3. Consider the large set of STS(9) from Example 1.9. Theorem 4.5 and Theorem 4.11 imply this set of announcements is an optimal $(3, 5, 1)$ strategy that is perfectly 1-secure against Cathy and informative for Bob.

In the case $n = 7$, there does not exist a large set of STS(7), so we cannot construct an optimal $(3, 3, 1)$-strategy. However, Example 4.2 provides us with an equitable strategy with $m = 10$ and $\gamma = 2$ that is informative for Bob and perfectly 1-secure against Cathy. This is because every announcement in this strategy is an STS(7) and every 3-subset occurs in exactly two announcements. Examples from the literature for this case typically only provide weak 1-security. Atkinson et al. [4] give a solution for the perfect 1-security case

that requires a much larger communication complexity m and also involves a complicated procedure in order to avoid card bias.

The following is an immediate consequence of Theorem 4.12 and Lemma 4.7.

Corollary 4.22. *Suppose* $(a, b, c) = (3, n-4, 1)$ *and suppose that Alice's strategy is equitable, informative for Bob, and perfectly 1-secure against Cathy. Then every announcement is a Steiner triple system.*

In fact, any $(a, b, a-2)$-strategy that is informative, equitable, and perfectly 1-secure also satisfies $c = 1$ (and hence $a = 3$).

Theorem 4.23. *Consider an* (a, b, c)*-deal such that* $a - c = 2$*. Suppose that Alice's strategy is equitable, informative for Bob, and perfectly 1-secure against Cathy. Then* $a = 3$ *and* $c = 1$*.*

Proof. Theorem 4.16 implies that every announcement is an $(a-1)$-design. Since $c \geq 1$, we have $a - 1 \geq a - c$, so we may apply Lemma 4.7. This implies $a - 1 = a - c$, so we have $c = 1$, as desired. □

We present an interesting example in the case $a = 4$, $c = 2$.

EXAMPLE 4.4. It was proven by Chouinard [**10**] that there is a large set of 2-$(13, 4, 1)$-designs. There are $\binom{11}{2} = 55$ designs in the large set. This yields a deterministic $(4, 7, 2)$-strategy that is informative for Bob. We can easily determine the security of the scheme against Cathy. Suppose that Alice's announcement is $\mathcal{A}_i$ and Cathy's hand is $H_C = \{y, z\}$. There is a unique block in $\mathcal{A}_i$ that contains the pair $\{y, z\}$, say $\{w, x, y, z\}$. There are three blocks that contain y but not z, and three blocks that contain z but not y. Since $\mathcal{A}_i$ contains 13 blocks, it follows that the set $\mathcal{P}(\{y, z\}, i)$ consists of six blocks. Within these six blocks, w and x occur three times, and every point in $X \backslash \{w, x, y, z\}$ occurs twice. Therefore, we have

$$\Pr[w \in H_A \mid H_C] = \Pr[x \in H_A \mid H_C] = \frac{1}{2}$$

and

$$\Pr[u \in H_A \mid H_C] = \frac{1}{3}$$

for all $u \in X \backslash \{w, x, y, z\}$. If a $(4, 7, 2)$-strategy were perfectly 1-secure against Cathy (which is impossible, in view of Theorem 4.23), we would have

$$\Pr[u \in H_A \mid H_C] = \frac{4}{11}$$

for all $u \in X \backslash H_C$.

We can generalize Theorem 4.23 and Corollary 4.22. That is, strategies that are equitable, informative for Bob, and perfectly $(a-c-1)$-secure against Cathy must satisfy $c=1$ and each announcement must be an $(a-1)$-$(n,a,1)$-design, also known as a Steiner system $S(a-1,a,n)$.

Theorem 4.24. *Consider an (a,b,c)-deal. Suppose that Alice's strategy is equitable, informative for Bob, and perfectly $(a-c-1)$-secure against Cathy. Then $c=1$.*

Proof. The proof is identical to the proof of Theorem 4.23. □

Corollary 4.25. *Consider an equitable $(a,b,1)$-strategy that is informative for Bob and perfectly $(a-2)$-secure against Cathy. Then every announcement is a Steiner system $S(a-1,a,n)$.*

Proof. The fact that every announcement is an $(a-1)$-design follows immediately from Theorem 4.16. To see that $\lambda=1$, we may apply Lemma 4.7. This is easy to see, however: since every $(a-1)$-subset occurs λ times, the fact that the strategy is informative for Bob implies $\lambda=1$. □

Example 4.5. The construction given in Example 4.6 is actually an example of a 2-equitable $(4,3,1)$-strategy that is informative for Bob and perfectly 2-secure against Cathy. (The fact that the scheme is perfectly 2-secure follows from Theorem 4.14.) As expected, each announcement is an $S(3,4,8)$.

The results in Corollaries 4.22 and 4.25 and Theorems 4.23 and 4.24 were first shown using a much more complicated proof technique in Swanson and Stinson [**75**]. In fact, we can use Theorem 4.16 and Lemma 4.7 to derive the following bound on the security parameter δ for perfectly δ-secure and informative strategies, which helps put the above results in context.

Corollary 4.26. *Suppose we have an equitable (a,b,c)-strategy that is perfectly δ-secure against Cathy and informative for Bob. Then $\delta \le a-2c$.*

Proof. If the strategy is perfectly δ-secure, then by Theorem 4.16, every announcement is a $(c+\delta)$-design. Now, if $c+\delta < a-c$ holds, then $\delta < a-2c$, as desired. If $c+\delta \ge a-c$, then since the strategy is informative for Bob, we can apply Lemma 4.7. This yields $c+\delta = a-c$, so we have $\delta = a-2c$ in this case. □

4.6.1. Construction methods and examples. Theorem 4.14 indicates that we can use t-designs to construct equitable strategies that are perfectly δ-secure against Cathy for some δ. In fact, so long as we use t-designs with $\lambda=1$ and $a-c \ge t$, such a strategy will also be informative for Bob (Corollary 4.4). This is a very interesting result, as we can use a single "starting design" to obtain equitable strategies that are informative for Bob and perfectly δ-secure against Cathy. We give a general method for this next. First we require some definitions.

Definition 4.3. Suppose that $\mathcal{D} = (X, \mathcal{B})$ is a t-(v, k, λ)-design. An *automorphism* of $\mathcal{D}$ is a permutation π of X such that π fixes the multiset $\mathcal{B}$. The collection of all automorphisms of $\mathcal{D}$ is denoted $\mathsf{Aut}(\mathcal{D})$; it is easy to see that $\mathsf{Aut}(\mathcal{D})$ is a subgroup of the symmetric group $S_{|X|}$.

Theorem 4.27. *Suppose $\mathcal{D} = (X, \mathcal{B})$ is a t-$(n, a, 1)$-design. Then there exists a γ-equitable $(a, n-a-c, c)$-strategy with m announcements that is informative for Bob and perfectly $(t-c)$-secure against Cathy for any choice of c such that $a-c \geq t$, where $m = n!/|\mathsf{Aut}(\mathcal{D})|$ and $\gamma = m/\binom{n-t}{a-t}$.*

Proof. Let the symmetric group S_n act on $\mathcal{D}$. We obtain a set of designs isomorphic to $\mathcal{D}$, which are the announcements in our strategy. Since each announcement is a t-$(n, a, 1)$-design, the resulting scheme is perfectly $(t-c)$-secure against Cathy by Theorem 4.14. Furthermore, since $a - c \geq t$ and $\lambda = 1$, no two blocks have more than $a - c - 1$ points in common, so Theorem 4.1 implies the scheme is informative for Bob.

The total number of designs m is equal to $n!/|\mathsf{Aut}(\mathcal{D})|$ (as this is the index of $\mathsf{Aut}(\mathcal{D})$ in S_n). To see that $\gamma = m/\binom{n-t}{a-t}$, consider a fixed t-subset A of X. Then in particular, there are $\binom{n-t}{a-t}$ possible blocks of size a that contain A. Now, every one of the m designs contains exactly one of these $\binom{n-t}{a-t}$ blocks, and these $\binom{n-t}{a-t}$ blocks occur equally often among the m designs. Thus, a given block B occurs in $m/\binom{n-t}{a-t}$ of the designs, as desired. □

Example 4.6. It is known that there is a 3-$(8, 4, 1)$-design having an automorphism group of order 1344. (See, for example, result 13 of Section 1.4 of Dembowski [**16**].) Theorem 4.27 thus yields a 6-equitable $(4, 3, 1)$-strategy with 30 announcements that is informative for Bob and perfectly 1-secure against Cathy. However, in this particular case, we can do better. Don Kreher (private communication) has found a set of ten 3-$(8, 4, 1)$-designs on a set of points $X = \{0, \ldots, 7\}$ such that every 4-subset of X occurs in exactly two of these designs. Therefore we have a 2-equitable $(4, 3, 1)$-strategy with ten announcements that is informative for Bob and perfectly 1-secure against Cathy. The set of 3-$(8, 4, 1)$-designs can be constructed as follows: Begin with a 3-$(8, 4, 1)$-design having the following set $\mathcal{A}_0$ of 14 blocks:

$$\{3,4,5,6\}, \{2,5,6,7\}, \{2,3,4,7\}, \{1,4,5,7\}, \{1,3,6,7\}, \{1,2,4,6\}, \{1,2,3,5\},$$
$$\{0,4,6,7\}, \{0,3,5,7\}, \{0,2,4,5\}, \{0,2,3,6\}, \{0,1,5,6\}, \{0,1,3,4\}, \{0,1,2,7\}.$$

Define the permutation $\pi = (0,1)(2)(3,4,6,7,5)$ and let π (and its powers) act on $\mathcal{A}_0$.

Remark 4.5. The technique described in Theorem 4.27 shows how to use a single "starting design" $\mathcal{D}$ on n points to construct a strategy that inherits its properties from $\mathcal{D}$. That is, the strategy obtained by letting the symmetric group S_n act on $\mathcal{D}$ will be informative and perfectly δ-secure if $\mathcal{D}$ is an informative announcement that satisfies Condition 2 of Definition 4.2 for the fixed announcement $\mathcal{D}$.

We can use the same method as in Theorem 4.27 to construct equitable (a, b, c) strategies that are perfectly δ-secure against Cathy, informative for Bob, and *allow Cathy to hold more than one card*. Such a solution to the generalized Russian cards problem has not yet been presented in the literature. We next give an infinite class of equitable and perfectly 1-secure strategies where Cathy holds two cards.

EXAMPLE 4.7. Consider the inversive plane with $q = 2^3$; this is a 3-$(65, 9, 1)$-design. (See Section 1.2.3.3.) The construction method in Theorem 4.27 yields an equitable $(9, 55, 1)$-strategy that is perfectly 2-secure against Cathy and informative for Bob and (more interestingly) a $(9, 54, 2)$-strategy that is perfectly 1-secure against Cathy and informative for Bob.

It is known that 3-$(q^2 + 1, q + 1, 1)$-designs (or inversive planes) exist whenever q is a prime power, as stated in Theorem 1.11. This gives us the following result.

Corollary 4.28. *There exists an equitable $(q + 1, q^2 - q - 2, 2)$-strategy that is informative for Bob and perfectly 1-secure against Cathy for every prime power $q \geq 4$.*

We now discuss some other constructions of strategies using results from design theory, including some applications of Remark 4.5.

It is clear that we can use any Steiner triple system, or 2-$(n, 3, 1)$-design, as a starting design to obtain an equitable $(3, n-4, 1)$-strategy that is informative for Bob and perfectly 1-secure against Cathy. As mentioned in Section 1.2.1, an STS(n) exists if and only if $n \equiv 1, 3 \bmod 6$, $n \geq 7$. We state this result in the following Corollary.

Corollary 4.29. *There exists an equitable $(3, n-4, 1)$-strategy for Alice that is informative for Bob and perfectly 1-secure against Cathy for any integer n such that $n \equiv 1, 3 \bmod 6$, $n \geq 7$.*

As discussed in Theorem 4.21, if we can construct a large set of 2-$(n, 3, 1)$-designs, this set forms an optimal strategy that is informative and perfectly 1-secure, and a large set of STS(n) exists whenever $n \equiv 1, 3 \bmod 6$ and $n > 7$. However, there are certain choices of n for which there is a particularly nice construction for a large set of STS(n), such that it would be easy for Alice and Bob to create this large set on their own. For details of this construction, we refer the reader to Section 1.2.3.2.

Two other types of designs that can be used to construct informative and perfectly 1-secure strategies where Cathy holds one card are hyperplanes in projective spaces and Hadamard designs. For a discussion of these constructions, we refer the reader to Section 1.2.3.1 and Section 1.2.3.4, respectively. We have the following results.

Corollary 4.30. *There exists an equitable $\left(\frac{q^d-1}{q-1}, q^d - 1, 1\right)$-strategy that is informative for Bob and perfectly 1-secure against Cathy, where $q \geq 2$ is a prime power and $d \geq 2$ is an integer.*

Proof. By Theorem 1.10, there exists a symmetric $\left(\frac{q^{d+1}-1}{q-1}, \frac{q^d-1}{q-1}, \frac{q^{d-1}-1}{q-1}\right)$-BIBD $\mathcal{D}$ for every prime power $q \geq 2$ and integer $d \geq 2$. The design $\mathcal{D}$ is a hyperplane in a projective space (or, in the case $d = 2$, a finite projective plane). Let the symmetric group S_n act on $\mathcal{D}$ as in the proof of Theorem 4.27, where $n = (q^{d+1} - 1)/(q - 1)$, to obtain Alice's strategy.

Lemma 4.11 immediately implies that this strategy is perfectly 1-secure against Cathy. To see that this strategy is informative, recall that the intersection of two blocks in a symmetric BIBD has size $\lambda = (q^{d-1} - 1)/(q - 1)$. It is easy to see that the strategy will be informative provided $a - c > \lambda$, which is the case here. □

Corollary 4.31. *There exists an equitable* $\left(\frac{q-1}{2}, \frac{q-1}{2}, 1\right)$*-strategy that is informative for Bob and perfectly* 1*-secure against Cathy, where* $q \equiv 3 \bmod 4$ *is an odd prime power.*

Proof. By Corollary 1.14, there exists a symmetric $\left(q, \frac{q-1}{2}, \frac{q-3}{4}\right)$-BIBD $\mathcal{D}$ for every odd prime power q such that $q \equiv 3 \bmod 4$. The design $\mathcal{D}$ is a Hadamard design. Let the symmetric group S_q act on $\mathcal{D}$ as in the proof of Theorem 4.27 to obtain Alice's strategy.

Lemma 4.11 immediately implies that this strategy is perfectly 1-secure against Cathy. To see that this strategy is informative, recall that the intersection of two blocks in a symmetric BIBD has size $\lambda = (q - 3)/4$. It is easy to see that the strategy will be informative provided $a - c > \lambda$, which is the case here. □

4.7. A Variant of the Russian Cards Problem

In this section, we consider a variation of the generalized Russian cards problem, in which we change the manner in which the cards are dealt. Our motivation for restricting the deal is to widen the solution space. Since the generalized Russian cards problem requires a suitable set of t-designs to maximize security against Cathy—and constructing t-designs for $t > 2$ is in general quite difficult—we explore certain types of deals where suitable constructions are more readily available. An added advantage of our deal restriction is that in this new framework, we can view Alice's hand as an a-tuple over an alphabet of size v. If Alice's hand represents a secret key, this variation is more in keeping with traditional key agreement schemes in cryptography, as typically secret keys are tuples rather than sets.

Suppose our deck X consists of $n = va$ cards, where v and a are positive integers such that $v > a$. Rather than allowing Alice, Bob, and Cathy to have any hand of the appropriate size, we first split the deck X into a piles, each of size v. Alice is given a hand H_A of a cards, such that she holds exactly one card from each pile. Cathy's hand H_C of c cards is assumed to contain no more than one card from each pile. The remainder of the deck becomes Bob's hand, H_B. Observe that we can use the same framework for this problem as for the original; we have only placed a limitation on the set of possible hands Alice, Bob, and Cathy might hold. The necessary modifications to the security definitions and the definition of an informative strategy are straightforward.

This variant admits a nice solution using *transversal designs*; we refer the reader to Section 1.2.4 for the relevant definitions and a discussion of these designs. In the context of a transversal design $\mathrm{TD}_\lambda(t, a, v)$, we can view the piles of cards as the groups $G_1, \ldots, G_a$ of the design. In this case, Alice's hand is a transversal and Cathy's hand is a partial transversal of $G_1, \ldots, G_a$. Note that Cathy therefore only considers transversals as possible hands for Alice. When we discuss weak (or perfect) δ-security, we are interested in the probability (from Cathy's point of view) that Alice holds partial transversals of order δ.

We first show Theorem 4.1 holds for this variant of the Russian cards problem:

Theorem 4.32. *The announcement $\mathcal{A}_i$ is informative for Bob if and only if there do not exist two distinct sets $H_A, H'_A \in \mathcal{A}_i$ such that $|H_A \cap H'_A| \geq a - c$.*

Proof. Suppose there exist two distinct sets $H_A, H'_A \in \mathcal{A}_i$ such that $|H_A \cap H'_A| \geq a - c$. We proceed by constructing a card deal consistent with the announcement $\mathcal{A}_i$ such that $\{H_A, H'_A\} \subseteq \mathcal{P}(H_B, i)$, which implies the announcement is not informative for Bob.

Write $|H_A \cap H'_A| = \ell$. Let Alice's hand be H_A, so it is possible for Alice to announce $\mathcal{A}_i$. Let Cathy's hand contain all the cards in H'_A that are not also contained in H_A; this is possible since $c \geq a - \ell$. Then Bob's hand H_B contains all the remaining cards. In particular, we have $H_B \cap (H_A \cup H'_A) = \emptyset$, so $\{H_A, H'_A\} \subseteq \mathcal{P}(H_B, i)$, as desired.

Conversely, suppose $\{H_A, H'_A\} \subseteq \mathcal{P}(H_B, i)$, where $H_A \neq H'_A$. Then $|H_A \cup H'_A| \leq n - b = a + c$, and hence $|H_A \cap H'_A| \geq a - c$. □

In light of Theorem 4.32, the following result is straightforward.

Theorem 4.33. *Consider an (a, b, c)-deal following the above rules and suppose that each announcement in an equitable (a, b, c)-strategy is a $\mathrm{TD}_1(t, a, v)$ satisfying $t \leq a - c$. Then the strategy is informative for Bob.*

We can use an argument similar to that of Theorem 4.3 to derive a lower bound on the size of Alice's announcement.

Theorem 4.34. *Suppose $a > c$ and there exists a strategy for Alice that is informative for Bob. Then the number of announcements m satisfies $m \geq v^c$.*

Proof. Fix a set of cards X' of size $a - c$, no two of which are from the same pile. There are v^c possible hands for Alice that contain X'. These hands must occur in different announcements, by Theorem 4.1 (which holds for this variation of the problem). Therefore $m \geq v^c$. □

As before, we refer to a strategy that meets this bound as *optimal*. We have the following result, which is similar Theorem 4.5.

Theorem 4.35. *Suppose that $a > c$. An optimal (a, b, c)-strategy for Alice that is informative for Bob is equivalent to a large set of $\mathrm{TD}_1(t, a, v)$, where $t = a - c$.*

Proof. Suppose there exists a large set of $\mathrm{TD}_1(a-c, a, v)$. Recall from Definition 1.37 that the set of all blocks sets (i.e., possible announcements) in this large set form a partition of the set of all transversals and that there are precisely v^c designs in such a set. Then it is easy to see that this immediately yields an optimal (a, b, c)-strategy for Alice that is informative for Bob.

Conversely, suppose there is an optimal (a, b, c)-strategy for Alice that is informative for Bob. We need to show that every announcement is a $\mathrm{TD}_1(a-c, a, v)$. As in the proof of Theorem 4.34, fix a set of cards X' of size $a-c$, no two of which are from the same pile. The v^c possible hands for Alice that contain X' must occur in different announcements. However, there are a total of v^c announcements, so every announcement must contain exactly one block that contains X'. □

The following result shows how transversal designs with arbitrary t can be used to achieve weak δ-security for permissible parameters $\delta \leq t-c$. As in Definition 1.36, for a transversal design $\mathrm{TD}_\lambda(t, a, v)$, say $(X, \mathcal{G}, \mathcal{B})$, and a partial transversal Y of $\mathcal{G}$, we let G_Y denote the set of groups of the transversal design that have nonempty intersection with the partial transversal Y.

Theorem 4.36. *Consider an (a, b, c)-deal following the above rules and suppose that each announcement in an equitable (a, b, c)-strategy is a $\mathrm{TD}_\lambda(t, a, v)$. Then the strategy is weakly $(t-c)$-secure against Cathy.*

Proof. Fix an announcement $\mathcal{A}_i$ for Alice. Suppose $\mathcal{A}_i$ is a $\mathrm{TD}_\lambda(t, a, v)$, say $(X, \mathcal{G}, \mathcal{B})$. Consider a possible hand H_C for Cathy. In particular, H_C is a partial transversal of the groups $G_1, \ldots, G_a \in \mathcal{G}$.

Since $c \leq t$, Theorem 1.17 implies there are

$$|\mathcal{P}(H_C, i)| = \lambda v^{t-c}(v-1)^c$$

blocks in $\mathcal{A}_i$ that do not contain any of the points of H_C.

Consider a partial transversal Y of order $\delta \leq t-c$. Since Y is not necessarily group disjoint from H_C, we must consider the number of groups which intersect both Y and H_C. In particular, the δ-subset Y never occurs with any other cards from $G_Y \cap G_{H_C}$, by definition of transversal designs.

Let $\ell = |G_{H_C} \backslash G_Y|$. That is, ℓ is the number of groups that do not intersect Y, but from which Cathy has cards. Write $z_1, \ldots, z_\ell$ for Cathy's cards from these ℓ groups. We wish to compute the number of blocks which contain all the points in Y but miss all of the points of H_C. This is the same as the number of blocks that contain all the points in Y but miss all the points in $\{z_1, \ldots, z_\ell\}$. Since $\ell + \delta \leq t$, by Theorem 1.17, we have $\lambda v^{t-\ell-\delta}(v-1)^\ell$ such blocks.

That is, a given set of points $x_1, \ldots, x_\delta \in X \backslash H_C$ that might be held by Alice is contained in precisely

$$|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_\delta \in H_A\}| = \lambda v^{t-\ell-\delta}(v-1)^\ell$$

of the blocks in $\mathcal{P}(H_C, i)$, where $\ell = \left|G_{H_C} \backslash G_{\{x_1,\ldots,x_\delta\}}\right|$.

Thus, for any partial transversal of δ distinct points $x_1, \ldots, x_\delta \in X \backslash H_C$, we have

$$\frac{|\{H_A \in \mathcal{P}(H_C, i) : x_1, \ldots, x_\delta \in H_A\}|}{|\mathcal{P}(H_C, i)|} = \frac{\lambda v^{t-\ell-\delta}(v-1)^\ell}{\lambda v^{t-c}(v-1)^c} = \frac{1}{v^{\delta+\ell-c}(v-1)^{c-\ell}},$$

so Condition 1 of Theorem 4.13 is satisfied. □

REMARK 4.6. We do not achieve perfect $(t-c)$-security in Theorem 4.36 because the number of hands of $\mathcal{P}(H_C, i)$ containing a given partial transversal Y of δ distinct points, where $\delta \leq t-c$, depends on $\ell = |G_{H_C} \backslash G_Y|$. In fact, we cannot expect to achieve better security than that of the construction given in Theorem 4.36 for this variant of the generalized Russian cards problem. This is because the rules for the deal imply that for each pile from which Cathy holds a card, Cathy knows that Alice holds one of the other $(v-1)$ cards, and for every other pile, Cathy knows only that Alice holds one of the other v cards.

As discussed in Section 1.2.4, large sets of transversal designs $\mathrm{TD}_\lambda(t, k, v)$ are easy to construct when you have a linear $\mathrm{TD}_\lambda(t, k, v)$ "starting design". As stated in Corollary 1.19, a linear $\mathrm{TD}_1(t, q, q)$ exists whenever the point set $X = (\mathbb{F}_q)^2$ and q is a prime power. The construction method for such a transversal design is simple; we refer the reader to the relevant discussion in Section 1.2.4 on Theorem 1.18 and Corollaries 1.19 and 1.20.

In particular, we can construct a linear $\mathrm{TD}_1(t, a, q)$ for a prime power $q \geq a$ by first constructing a $\mathrm{TD}_1(t, q, q)$ and then (if necessary) deleting $q-a$ groups. This yields a wide range of informative and weakly $(t-c)$-secure $(a, n-a-c, c)$-strategies for card decks of size $n = aq$ and any choice of c satisfying $a - c \geq t$. If we take $t = a - c$, these strategies are optimal. We summarize this result in the following theorem.

Theorem 4.37. *Consider the above variant of the generalized Russian cards problem. Let q be a prime power such that $q \geq a$. Then there exists an equitable $(a, aq-a-c, c)$-strategy that is optimal, informative for Bob, and weakly $(a-2c)$-secure against Cathy.*

4.8. Discussion and Comparison with Related Work

As mentioned in Section 4.1, there are have been many papers studying the Russian cards problem and generalizations of it. Some are interested in card deal protocols that allow players to agree on a common secret without a given eavesdropper being able to determine this secret value. This area of research is especially interesting in terms of possible applications to key generation [**2**, **3**, **31**–**35**, **47**, **51**]. Fischer and Wright [**32**–**35**] have been especially prolific on this topic, following the paper by Fischer et al. [**31**], which began the investigation into secret bit transmission protocols based on card deals amongst

three players. A useful concept for secret bit transmission is that of a *key set*, which is a set K of two cards held by two different players A and B. A key set is said to be *hidden* if, from Eve's point of view, either player is equally likely to hold a given card in K [31].

Fischer and Wright [32] consider multi-party secret key exchange, where a deck of cards is split among $k+1$ participants in a specified manner, with one participant singled out as the eavesdropper Eve. The idea is that Eve, who is computationally unlimited, should not be privy to a secret key established among the k other players using public announcements; the protocols established build on the notion of hidden key sets. Fischer and Wright extended this work in later papers [34], including using game-theoretic techniques to analyze strategies among key set protocols [33], and providing a formal security model for the problem [35]. Mizuki et al. [51] and Koizumi et al. [47] continued work on key exchange using key set protocols.

Other papers are concerned with analyzing variations of the problem using epistemic logic [15, 19–21]. van Ditmarsch [20, 21] presents a comprehensive analysis of the original Russian cards problem (with parameters $(3, 3, 1)$), including some discussion on removing the requirement for complete public knowledge of the protocol. That is, van Ditmarsch considers cases where it is unknown whether or not the two players actually know each other's hands and that the protocol is finished. As this is not in keeping with Kerckhoffs' principle, we do not consider this case. van Ditmarsch et al. [19] implement a protocol and compare results using various epistemic model checkers. Finally, Cyriac and Krishnan [15] use dynamic epistemic logic to show that there exists no single-announcement solution for the original Russian cards problem, and that if the adversary has a sufficiently large number of cards, no two-announcement solution exists for the generalized Russian cards problem of the form (k, k, ℓ).

Recently, Duan and Yang [30] and He and Duan [41] consider a special generalization, with $n-1$ players each dealt n cards, and one player (the intruder) dealt one card; the authors give an algorithm by which a dealer, acting as a trusted third party, can construct announcements for each player.

Of more relevance to us is the recent research that takes a combinatorial approach [1–4, 12], on which we now focus. Albert et al. [1] consider the card problem from both epistemic logic and combinatorial perspectives, establishing axioms CA1, CA2, and CA3 that are roughly equivalent to our requirements for a protocol to be informative and weakly 1-secure in the γ-equitable case. The difference is that the authors [1] treat security on the announcement level; that is, they identify various announcements as *good* if the relevant properties hold for any possible hand for Alice *in the given announcement*. No assumption is made that, for every possible hand for Alice, an announcement is defined, or that a good announcement even exists. Our definitions, on the other hand, require that Alice have a (secure) announcement for every possible hand $H_A \in \binom{X}{a}$. In particular, we argue that it is not possible to formally define or discuss the security of a scheme using definitions that focus on individual announcements.

Albert et al. [1] present several useful results, some of which we have cited in this work, on the relationships between the parameters a and c, and b and c, as well as bounds on the minimum and maximum number of hands in a good announcement. The focus is on the level of announcements throughout; the authors argue that, to minimize information gained by Cathy, the size of the announcement should be maximized. Moreover, the authors show good announcements exist for some special cases, including using block designs for the case $(a, 2, 1)$, when $a \equiv 0, 4 \bmod 6$ (corresponding to the Steiner triple systems), and using Singer difference sets for the case (a, b, c), where a and c are given, and b is sufficiently large. A few other small cases are also given.

Atkinson et al. [4] extend these notions to include a new axiom, CA4, which roughly corresponds to our notion of perfect 1-security. That is, the authors recognize the possibility of card occurrence bias in a good announcement, which gives Cathy an advantage in guessing Alice's hand. Axiom CA4 introduces the requirement that, in the set of hands Cathy knows are possible for Alice, each card Cathy does not hold occurs a constant number of times. In this setting, the authors use binary designs to construct a good announcement (also satisfying CA4) for parameters of the form $\left(2^{k-1}, 2^{k-1} - 1, 1\right)$, where $k \geq 3$. Atkinson et al. also consider the problem of unbiasing an announcement by applying a protocol that takes the existence of bias into account. An example of two possible methods for achieving this are given for the parameter set $(3, 3, 1)$. Our approach is much simpler and yields nice solutions for the $(3, 3, 1)$ case. In particular, we require fewer announcements and thereby less communication complexity. We remark that this work by Atkinson et al. [4] is the only work treating security notions stronger than weak 1-security of which we are aware, other than our paper [75] and subsequent work by Cordón-Franco et al. [14].

Albert et al. [2,3] investigate the problem of Alice and Bob communicating their hands in light of two different security goals with respect to Cathy. The first goal, that Cathy does not learn the fate of any given card with certainty, or that the protocol itself is *card safe*, is similar to our notion of weak 1-security. The analysis includes a sum announcement protocol which is card safe for the case $(k, k, 1)$, where $k \geq 3$; that is, both players announce the sum of their cards modulo $2k + 1$.

The second, more relaxed goal is that Cathy does not learn the entire card deal (but may determine ownership of some subset of the cards), or that the protocol is *state safe*. In this setting, a state informative and state safe protocol consisting of three announcements is given for the card deal $(2, 2, 1)$, and the authors observe that no shorter protocol exists.

The main idea behind a state safe protocol is that this relaxed security condition may be sufficient for Alice and Bob to establish a secret *bit*. In particular, Albert et al. delve into the realm of epistemic logic and consider a proposition p such that it is public knowledge that Alice and Bob know the value of p, but Cathy does not know the value of p. Here the main concern is with propositions p describing the card deal, and the basic idea is that a protocol which is *state informative* is also *bit informative with respect to* p. Moreover, in this context, *state safe* and *bit safe* are equivalent; the proof is given by Albert et al. [2].

Albert et al. [2, 3] also pose the question of whether the existence of a bit informative protocol implies the existence of a state informative protocol and conjecture that the answer is affirmative. A discussion of of parameters (a, b, c) for which bit exchange protocols exist is included (namely a, $b > c$ or $a > b = c > 0$ or $b > a = c > 0$). The protocols themselves are closely related to the work of Fischer et al. [31] and Fischer and Wright [35], and are based on the simple observation that if $c = 0$, Alice and Bob know the card deal already, and therefore can establish a secret bit relative to whether "Alice holds card x" for any card x. If $c \neq 0$, then Alice can announce some subset D of cards, and hope that Bob can honestly respond "I hold all but one of D". In this case, Alice and Bob both know which card from D is held by Alice, thereby establishing the secret bit. If Bob does not hold all but one card in D, then Alice and Bob can throw the cards in D away until they are dealing with the situation $c = 0$ or they otherwise know the card deal (which can happen when $c = 1$, depending on how D is chosen).

Cordón-Franco et al. [12] focus on the case c = 1, and present a protocol in which Alice and Bob announce the sum of their hands modulo a given (public) integer. The authors deal with the case of the modulus being either n (the size of the deck) or the least prime p larger than n, and show that, by choosing one of these protocols as appropriate, deals of the form $(a, b, 1)$ (where $a, b \geq 3$) are secure (in the weak 1-secure sense) and informative. That is, Alice and Bob learn each other's cards, but afterwards Cathy does not know with certainty if Alice (or Bob) holds any particular individual card not in Cathy's hand. This approach yields strategies for a wide variety of parameters, but these strategies satisfy only weak 1-security.

In addition, there has been recent work [13, 22] in which protocols consisting of more than once announcement by Alice and Bob are considered, which is a generalization of the problem which we consider here. van Ditmarsch and Soler-Toscano [22] show that no good announcement exists for card deals of the form $(4, 4, 2)$ using bounds from Albert et al. [1]. The authors instead give an interactive protocol that requires at least three rounds of communication in order for Alice and Bob to learn each other's hands; their protocol uses combinatorial designs to determine the initial announcement by Alice and the protocol analysis is done using epistemic logic.

Cordón-Franco et al. [13] consider four-step solutions for the generalized Russian cards problem with parameters (a, b, c) such that $c > a$. Although Cordón-Franco et al. [13] present a "protocol", their solution is not a protocol in the typical sense of the word, as it is unclear if the protocol is executable or not. The authors demonstrate the existence of a necessary construction for their protocol when the card deal parameters satisfy specific conditions, but do not address the feasibility of finding such constructions in practice. In particular, the security of the protocol itself relies heavily on the ability of the players to pick such a construction uniformly at random from all possible constructions. Since it is unclear if this is feasible, the protocol is questionable, albeit theoretically interesting in that it attempts to treat cases where $c > a$.

Cordón-Franco et al. [14] further elaborate on protocols of length two and our notion of weak k-security. The authors present a geometric protocol based on hyperplanes that yields informative and weakly k-secure equitable (a, b, c)-strategies for appropriate parameters. In particular, this protocol allows Cathy to hold more than one card.

This chapter draws material from Swanson and Stinson [75], but we greatly extend and simplify the results. In particular, our main result establishing an equivalence between equitable perfectly δ-secure strategies and suitable sets of $(c + \delta)$-designs with point set X and block size a, stated in Theorem 4.18, is new, as are most of the auxiliary results presented in Section 4.5. We show new results connecting informative strategies and t-designs in Section 4.4, which together with the new material in Section 4.5, allows us to simplify the proofs for results connecting certain types of perfectly δ-secure deals and Steiner systems, originally shown in Swanson and Stinson [75].

Most of the material on construction methods and examples for simultaneously informative and secure strategies, presented in Section 4.6.1, is new, and the construction technique using a "starting design" is a generalization of the technique given by Swanson and Stinson [75]. This generalized construction technique allows us to answer in the affirmative the open question on the existence of perfectly secure and informative strategies for deals in which Cathy holds more than one card. The equivalence we have shown between such strategies and t-designs, together with the difficulty of constructing t-designs for $t > 2$ in general, motivated the proposed variant of the generalized Russian cards problem, discussed in Section 4.7. This variant allows us to take advantage of transversal designs, which are in general much easier to construct than t-designs.

4.9. Concluding Remarks and Future Work

We have presented the first formal mathematical presentation of the generalized Russian cards problem, and have provided rigorous security definitions that capture both basic and extended versions of weak and perfect security notions. Using a combinatorial approach, we are able to give a nice characterization of informative strategies having optimal communication complexity, namely the set of announcements must be equivalent to a large set of t-$(n, a, 1)$-designs, where $t = a - c$. We also characterize γ-equitable strategies that are perfectly δ-secure for some δ, showing an equivalence between such a strategy and a set of $(c + \delta)$-designs on n points with block size a, where this set must satisfy the additional property that every a-subset of X occurs in precisely γ of these designs.

Moreover, we show how to use a "starting" t-$(n, a, 1)$-design to construct equitable (a, b, c)-strategies that are informative and perfectly $(t - c)$-secure against Cathy for any choice of c satisfying $a - c \geq t$. In particular, this indicates that if an appropriate t-design exists, it is possible to achieve perfect security for deals where Cathy holds more than one card. We present an example construction, based on inversive planes, for $(q+1, q^2-q-2, 2)$-strategies which are perfectly 1-secure against Cathy and informative for Bob, where q is a prime power.

In addition, we discuss a variation of the Russian cards problem which admits nice solutions using transversal designs. The variant changes the manner in which the cards are dealt, but the resulting problem can be solved using large sets of transversal designs with $\lambda = 1$ and arbitrary t, which are easy to construct. In particular, this solution is optimal in terms of the number of announcements and provides the strongest possible security for appropriate parameters. That is, for card decks of size aq, where $q \geq a$ is a prime power, we achieve $(a, aq - a - c, c)$-strategies that are optimal, informative for Bob, and weakly $(a - 2c)$-secure against Cathy.

There are many open problems in the area, especially for deals with $c > 1$. Given the general difficulty of constructing t-designs for $t > 2$ and $\lambda = 1$, we see that constructing perfectly δ-secure and informative strategies for $c > 1$ is a difficult combinatorial problem. A more promising direction for the case $c > 1$ may be strategies that are weakly δ-secure for $\delta > 1$, a concept we introduced and which has received some attention in current literature [**14**]. In particular, further characterizing such strategies using combinatorial notions might prove informative.

Bibliography

[1] Albert, M.H., Atkinson, M.D., van Ditmarsch, H.P., Handley, C., Aldred, R.E.L.: Safe communication for card players by combinatorial designs for two-step protocols. Australasian Journal of Combinatorics 33, 33–46 (2005) 97, 101, 105, 111, 112, 121, 122, 123

[2] Albert, M.H., Cordón-Franco, A., van Ditmarsch, H.P., Fernández-Duque, D., Joosten, J.J., Soler-Toscano, F.: Secure communication of local states in multi-agent systems. http://personal.us.es/hvd/newpubs/fLiSsecretl.pdf (2010), extended version of [**3**] 120, 121, 122, 123

[3] Albert, M.H., Cordón-Franco, A., van Ditmarsch, H.P., Fernández-Duque, D., Joosten, J.J., Soler-Toscano, F.: Secure communication of local states in interpreted systems. In: Abraham, A., Corchado, J.M., Rodríguez-González, S., Santana, J.F.D.P. (eds.) Distributed Computing and Artificial Intelligence (DCAI 2011). Advances in Soft Computing, vol. 91, pp. 117–124. Springer (2011) 96, 97, 120, 121, 122, 123, 126

[4] Atkinson, M.D., van Ditmarsch, H.P., Roehling, S.: Avoiding bias in cards cryptography. Australasian Journal of Combinatorics 44, 3–18 (2009) 97, 105, 112, 121, 122

[5] Bras-Amorós, M., Stokes, K., Greferath, M.: Problems related to combinatorial configurations with applications to P2P-user private information retrieval. In: Mathematical Theory of Networks and Systems (MTNS 2010). pp. 1267–1271 (2010) 93

[6] Brickell, E.F., Stinson, D.R.: Authentication codes with multiple arbiters (extended abstract). In: Günther, C.G. (ed.) Advances in Cryptology – EUROCRYPT '88. Lecture Notes in Computer Science, vol. 330, pp. 51–55. Springer (1988) 21

[7] Castellà-Roca, J., Viejo, A., Herrera-Joancomartí, J.: Preserving user's privacy in web search engines. Computer Communications 32(13–14), 1541–1551 (2009) 94

[8] Chaum, D., Roijakkers, S.: Unconditionally secure digital signatures. In: Menezes and Vanstone [**50**], pp. 206–214 21, 57

[9] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Foundations of Computer Science (FOCS '95). p. 41. IEEE Computer Society (1995) 61

[10] Chouinard II, L.G.: Partitions of the 4-subsets of a 13-set into disjoint projective planes. Discrete Mathematics 45(2–3), 297–300 (1983) 113

[11] Colbourn, C.J., Dinitz, J.H.: The CRC Handbook of Combinatorial Designs. Chapman & Hall/CRC, 2nd edn. (2006) 9, 13

[12] Cordón-Franco, A., van Ditmarsch, H.P., Fernández-Duque, D., Joosten, J.J., Soler-Toscano, F.: A secure additive protocol for card players. Australasian Journal of Combinatorics 54, 163–176 (2012) 97, 121, 123

[13] Cordón-Franco, A., van Ditmarsch, H.P., Fernández-Duque, D., Soler-Toscano, F.: A colouring protocol for the generalized Russian cards problem. CoRR abs/1207.5216 (2013) 97, 123

[14] Cordón-Franco, A., van Ditmarsch, H.P., Fernández-Duque, D., Soler-Toscano, F.: A geometric protocol for cryptography with cards. CoRR abs/1301.4289 (2013) 97, 122, 124, 125

[15] Cyriac, A., Krishnan, K.M.: Lower bound for the communication complexity of the Russian cards problem. CoRR abs/0805.1974 (2008) 96, 121

[16] Dembowski, P.: Finite Geometries. Springer-Verlag, New York (1968) 115

[17] Desmedt, Y., Frankel, Y., Yung, M.: Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback. In: IEEE International Conference on Computer Communications (IEEE INFOCOM '92). vol. 3, pp. 2045–2054. IEEE Computer Society Press (1992) 21
[18] Desmedt, Y., Yung, M.: Abritrated unconditionally secure authentication can be unconditionally protected against arbiter's attacks (extended abstract). In: Menezes and Vanstone [**50**], pp. 177–188 21
[19] van Ditmarsch, H.P., van der Hoek, W., van der Meyden, R., Ruan, J.: Model checking Russian cards. Electronic Notes in Theoretical Computer Science 149(2), 105–123 (2006) 96, 121
[20] van Ditmarsch, H.P.: The Russian cards problem. Studia Logica 75(1), 31–62 (2003) 96, 121
[21] van Ditmarsch, H.P.: The case of the hidden hand. Journal of Applied Non-Classical Logics 15(4), 437–452 (2005) 96, 121
[22] van Ditmarsch, H.P., Soler-Toscano, F.: Three steps. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) Computational Logic in Multi-Agent Systems (CLIMA XII). Lecture Notes in Computer Science, vol. 6814, pp. 41–57. Springer (2011) 97, 123
[23] Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-insulated public key cryptosystems. In: Knudsen [**46**], pp. 65–82 22, 44, 48, 58
[24] Dodis, Y., Katz, J., Xu, S., Yung, M.: Strong key-insulated signature schemes. In: Desmedt, Y. (ed.) Public Key Cryptography (PKC 2003). Lecture Notes in Computer Science, vol. 2567, pp. 130–144. Springer (2003) 22, 45, 58
[25] Domingo-Ferrer, J., Solanas, A., Castellà-Roca, J.: $h(k)$-private information retrieval from privacy-uncooperative queryable databases. Journal of Online Information Review 33(4), 720–744 (2009) 94
[26] Domingo-Ferrer, J.: Coprivacy: Towards a theory of sustainable privacy. In: Domingo-Ferrer, J., Magkos, E. (eds.) Privacy in Statistical Databases (PSD 2010). Lecture Notes in Computer Science, vol. 6344, pp. 258–268. Springer (2010) 94
[27] Domingo-Ferrer, J., Bras-Amorós, M.: Peer-to-peer private information retrieval. In: Domingo-Ferrer, J., Saygin, Y. (eds.) Privacy in Statistical Databases (PSD 2008). Lecture Notes in Computer Science, vol. 5262, pp. 315–323. Springer (2008) 62, 66, 93
[28] Domingo-Ferrer, J., Bras-Amorós, M., Wu, Q., Manjón, J.A.: User-private information retrieval based on a peer-to-peer community. Data & Knowledge Engineering 68(11), 1237–1252 (2009) 61, 62, 66, 67, 93
[29] Domingo-Ferrer, J., González-Nicolás, Ú.: Rational behavior in peer-to-peer profile obfuscation for anonymous keyword search. Information Sciences 185(1), 191–204 (2012) 95
[30] Duan, Z., Yang, C.: Unconditional secure communication: a Russian cards protocol. Journal of Combinatorial Optimization 19, 501–530 (2010) 96, 121
[31] Fischer, M.J., Paterson, M.S., Rackoff, C.: Secret bit transmission using a random deal of cards. In: Discrete Mathematics and Theoretical Computer Science. DIMACS, vol. 2, pp. 173–181. American Mathematical Society (1991) 96, 120, 121, 123
[32] Fischer, M.J., Wright, R.N.: Multiparty secret key exchange using a random deal of cards. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO '91. Lecture Notes in Computer Science, vol. 576, pp. 141–155. Springer (1991) 96, 120, 121
[33] Fischer, M.J., Wright, R.N.: An application of game theoretic techniques to cryptography. In: Discrete Mathematics and Theoretical Computer Science. DIMACS, vol. 13, pp. 99–118. American Mathematical Society (1993) 96, 120, 121
[34] Fischer, M.J., Wright, R.N.: An efficient protocol for unconditionally secure secret key exchange. In: ACM-SIAM Symposium on Discrete algorithms (SODA '93). pp. 475–483. Society for Industrial and Applied Mathematics (1993) 96, 120, 121
[35] Fischer, M.J., Wright, R.N.: Bounds on secret key exchange using a random deal of cards. Journal of Cryptology 9(2), 71–99 (1996) 96, 120, 121, 123

[36] Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990) 73
[37] GoogleSharing: Keep your search history yours. http://www.googlesharing.net/googlesharing/, accessed February 1, 2013 94
[38] Hanaoka, G., Shikata, J., Zheng, Y., Imai, H.: Unconditionally secure digital signature schemes admitting transferability. In: Okamoto, T. (ed.) Advances in Cryptology – ASIACRYPT 2000. Lecture Notes in Computer Science, vol. 1976, pp. 130–142. Springer (2000) 20, 21, 22, 23, 37, 44, 45, 49, 57, 58, 60
[39] Hanaoka, G., Shikata, J., Zheng, Y., Imai, H.: Efficient and unconditionally secure digital signatures and a security analysis of a multireceiver authentication code. In: Naccache, D., Paillier, P. (eds.) Public Key Cryptography (PKC 2002). Lecture Notes in Computer Science, vol. 2274, pp. 64–79. Springer (2002) 20, 21, 22, 57
[40] Hara, Y., Seito, T., Shikata, J., Matsumoto, T.: Unconditionally secure blind signatures. In: Desmedt, Y. (ed.) Information Theoretic Security (ICITS 2007), Lecture Notes in Computer Science, vol. 4883, pp. 23–43. Springer (2009) 20, 21, 31, 57, 60
[41] He, J., Duan, Z.: Public communication based on Russian cards protocol: A case study. In: Wang, W., Zhu, X., Du, D.Z. (eds.) Combinatorial Optimization and Applications (COCOA 2011). Lecture Notes in Computer Science, vol. 6831, pp. 192–206. Springer (2011) 96, 121
[42] Howe, D., Nissenbaum, H.: TrackMeNot: Resisting surveillance in web search. In: Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society, pp. 417–436. Oxford University Press (2009) 94
[43] Johansson, T.: On the construction of perfect authentication codes that permit arbitration. Lecture Notes in Computer Science, vol. 773, pp. 343–354. Springer (1993) 21
[44] Johansson, T.: Further results on asymmetric authentication schemes. Information and Computation 151(1–2), 100–133 (1999) 21, 57
[45] Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Cryptography and Network Security, Chapman & Hall/CRC (2008) 3
[46] Knudsen, L.R. (ed.): Advances in Cryptology – EUROCRYPT 2002, Lecture Notes in Computer Science, vol. 2332. Springer (2002) 127, 129
[47] Koizumi, K., Mizuki, T., Nishizeki, T.: Necessary and sufficient numbers of cards for the transformation protocol. In: Chwa, K.Y., Munro, J.I. (eds.) Computing and Combinatorics (COCOON 2004). Lecture Notes in Computer Science, vol. 3106, pp. 92–101. Springer (2004) 96, 120, 121
[48] Lee, J., Stinson, D.R.: A combinatorial approach to key predistribution for distributed sensor networks. In: IEEE Wireless Communications and Networking Conference (WCNC 2005). pp. 1200–1205. IEEE Computer Society Press (2005) 69
[49] Mathon, R., Street, A.P.: Partitions of sets of designs on seven, eight and nine points. Journal of Statistical Planning and Inference 58(1), 135–150 (1997) 12
[50] Menezes, A., Vanstone, S.A. (eds.): Advances in Cryptology – CRYPTO '90, Lecture Notes in Computer Science, vol. 537. Springer (1991) 126, 127
[51] Mizuki, T., Shizuya, H., Nishizeki, T.: A complete characterization of a family of key exchange protocols. International Journal of Information Security 1(2), 131–142 (2002) 96, 120, 121
[52] Motwani, R., Raghavan, P.: Randomized Algorithms, chap. Tail Inequalities, pp. 67–73. Cambridge University Press (1995) 81
[53] Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf (2010), v0.34 61

[54] Raymond, J.F.: Traffic analysis: Protocols, attacks, design issues, and open problems. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies, Lecture Notes in Computer Science, vol. 2009, pp. 10–29. Springer Berlin Heidelberg (2001) 68
[55] Rebollo-Monedero, D., Forné, J., Domingo-Ferrer, J.: Query profile obfuscation by means of optimal query exchange between users. IEEE Transactions on Dependable and Secure Computing 9(5), 641–654 (2012) 95
[56] Reiter, M.K., Rubin, A.D.: Crowds: anonymity for web transactions. ACM Transactions on Information and System Security (TISSEC) 1(1), 66–92 (1998) 81, 94
[57] Roeder, T., Pass, R., Schneider, F.: Multi-verifier signatures. Journal of Cryptology 25, 310–348 (2012) 57
[58] Safavi-Naini, R., McAven, L., Yung, M.: General group authentication codes and their relation to "unconditionally-secure signatures". In: Bao, F., Deng, R.H., Zhou, J. (eds.) Public Key Cryptography (PKC 2004). Lecture Notes in Computer Science, vol. 2947, pp. 231–247. Springer (2004) 20, 21, 22, 31, 32, 57
[59] Safavi-Naini, R., Wang, H.: Broadcast authentication in group communication. In: Lam, K.Y., Okamoto, E., Xing, C. (eds.) Advances in Cryptology – ASIACRYPT '99. Lecture Notes in Computer Science, vol. 1716, pp. 399–411. Springer (1999) 21
[60] Sánchez, D., Castellà-Roca, J., Viejo, A.: Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines. Information Sciences 218, 17–30 (2013) 94
[61] Schreiber, S.: Covering all triples on n marks by disjoint steiner systems. Journal of Combinatorial Theory, Series A 15(3), 347–350 (1973) 13
[62] Seito, T., Shikata, J.: Information-theoretically secure key-insulated key-agreement. In: IEEE Information Theory Workshop (ITW 2011). pp. 287–291. IEEE Computer Society Press (2011) 22, 45, 58, 60
[63] Seito, T., Aikawa, T., Shikata, J., Matsumoto, T.: Information-theoretically secure key-insulated multireceiver authentication codes. In: Bernstein, D.J., Lange, T. (eds.) Progress in Cryptology – AFRICACRYPT 2010. Lecture Notes in Computer Science, vol. 6055, pp. 148–165. Springer (2010) 22, 45, 49, 58, 60
[64] Shannon, C.E.: Communication theory of secrecy systems. Bell system technical journal 28(4), 656–715 (1949) 3
[65] Shikata, J., Hanaoka, G., Zheng, Y., Imai, H.: Security notions for unconditionally secure signature schemes. In: Knudsen [**46**], pp. 434–449 20, 21, 22, 31, 32, 44, 57, 58, 59, 60
[66] Simmons, G.J.: Message authentication with arbitration of transmitter/receiver disputes. In: Chaum, D., Price, W.L. (eds.) Advances in Cryptology – EUROCRYPT '87. Lecture Notes in Computer Science, vol. 304, pp. 151–165. Springer (1987) 21
[67] Simmons, G.J.: A cartesian product construction for unconditionally secure authentication codes that permit arbitration. Journal of Cryptology 2, 77–104 (1990) 21, 59
[68] Stinson, D.R.: Combinatorial Designs: Constructions and Analysis. Springer-Verlag (2003) 9, 13
[69] Stinson, D.R.: Cryptography: Theory and Practice. Discrete Mathematics and its Applications, Chapman & Hall/CRC, 3rd edn. (2006) 3
[70] Stokes, K.: Combinatorial structures for anonymous database search. Ph.D. dissertation, Universitat Rovira i Virgili, Tarragona (2011) 93
[71] Stokes, K., Bras-Amorós, M.: Optimal configurations for peer-to-peer user-private information retrieval. Computers & Mathematics with Applications 59(4), 1568–1577 (2010) 62, 66, 68, 93
[72] Stokes, K., Bras-Amorós, M.: Combinatorial structures for an anonymous data search protocol. In: Workshop on Computational Security. Centre de Recerca Matemàtica (CRM), Barcelona, Spain (2011) 68, 69, 93, 94

[73] Stokes, K., Bras-Amorós, M.: On query self-submission in peer-to-peer user-private information retrieval. In: Truta, T.M., Xiong, L., Fotouhi, F., Orsborn, K., Stefanova, S. (eds.) Privacy and Anonymity in Information Society (PAIS '11). pp. 7:1–7:5. ACM (2011) 62, 66, 67, 68, 80, 93
[74] Stokes, K., Farràs, O.: Linear spaces and transversal designs: k-anonymous combinatorial configurations for anonymous database search notes. Designs, Codes and Cryptography pp. 1–22 (2012) 94, 95
[75] Swanson, C.M., Stinson, D.R.: Combinatorial solutions providing improved security for the generalized Russian cards problem. Designs, Codes and Cryptography pp. 1–23 (2012) 96, 114, 122, 124
[76] Swanson, C.M., Stinson, D.R.: Unconditionally secure signature schemes revisited. In: Fehr, S. (ed.) Information Theoretic Security (ICITS 2011). Lecture Notes in Computer Science, vol. 6673, pp. 100–116. Springer (2011) 20
[77] Swanson, C.M., Stinson, D.R.: Extended combinatorial constructions for peer-to-peer user-private information retrieval. Advances in Mathematics of Communications 6, 479–497 (2012) 61, 93, 94
[78] Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557–570 (2002) 93
[79] Toubiana, V., Subramanian, L., Nissenbaum, H.: TrackMeNot: Enhancing the privacy of web search. CoRR abs/1109.4677 (2011) 94
[80] Viejo, A., Castellà-Roca, J.: Using social networks to distort users' profiles generated by web search engines. Computer Networks 54(9), 1343–1357 (2010) 94
[81] Wilson, R.: Some partitions of all triples into steiner triple systems. In: Berge, C., Ray-Chaudhuri, D. (eds.) Hypergraph Seminar, Lecture Notes in Mathematics, vol. 411, pp. 267–277. Springer (1974), 10.1007/BFb0066198 13, 14
[82] Wright, M., Adler, M., Levine, B.N., Shields, C.: An analysis of the degradation of anonymous protocols. In: Network and Distributed System Security Symposium (NDSS 2002). The Internet Society (2002) 81, 94
[83] Wright, M.K., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: An analysis of a threat to anonymous communications systems. ACM Transactions on Information and System Security (TISSEC) 7(4), 489–522 (2004) 81, 94

APPENDIX A

Analysis of USS Constructions

We need the following lemmas:

Lemma A.1. *Let $n \in \mathbb{N}$ and consider the set of $n+1$ vectors*

$$R = \{\vec{r_i} = (r_{i,1}, \ldots, r_{i,n}) \in (\mathbb{F}_q)^n : i = 1, \ldots, n+1\}.$$

If the set of vectors $\{\vec{r_i}' = (1, r_{i,1}, \ldots, r_{i,n}) \in (\mathbb{F}_q)^{n+1} : i = 1, \ldots, n+1\}$ form a linearly independent set, then there exists a subset $R' \subset R$ of linearly independent vectors of size n.

Proof. Consider the matrix

$$M = \begin{pmatrix} 1 & r_{1,1} & r_{1,2} & \ldots & r_{1,n} \\ 1 & r_{2,1} & r_{2,2} & \ldots & r_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & r_{n+1,1} & r_{n+1,2} & & r_{n+1,n} \end{pmatrix}.$$

Let M_{ij} denote the (i,j) minor matrix of M. Then calculating the determinant of M by expansion along the first column, we have

$$\det(M) = \sum_{i=1}^{n+1} (-1)^{i+1} \det(M_{i,1}). \tag{22}$$

Recall M is invertible, so $\det(M) \neq 0$. Thus (22) implies $\det(M_{k,1}) \neq 0$ for some $k \in \{1, \ldots n\}$. We conclude that the matrix $M_{k,1}$ is invertible, so the desired subset R' exists. □

Lemma A.2. *Let $n \in \mathbb{N}$ and let F and F' be polynomials in $y_1, \ldots, y_n$ of the form $a_0 + \sum_{i=1}^{n} a_i y_i$ over $\mathbb{F}_q$. Suppose F' and F agree on the $n+1$ vectors*

$$R = \{\vec{r_i} = (r_{i,1}, \ldots, r_{i,n}) \in (\mathbb{F}_q)^n : i = 1, \ldots, n+1\}.$$

If the set of vectors $\{\vec{r_i}' = (1, r_{i,1}, \ldots, r_{i,n}) \in (\mathbb{F}_q)^{n+1} : i = 1, \ldots, n+1\}$ form a linearly independent set, then $F' = F$.

Proof. Define linear homogeneous polynomials G, $G' \in \mathbb{F}_q[y_0, \ldots, y_n]$ such that

$$G(y_0, \ldots, y_n)|_{y_0=1} = F(y_1, \ldots, y_n)$$

and

$$G'(y_0, \ldots, y_n)|_{y_0=1} = F'(y_1, \ldots, y_n).$$

We have $(G - G')(1, r_{i,1}, \ldots, r_{i,n}) = 0$ for $1 \leq i \leq n+1$. In particular, this forms a homogeneous linear system of $n+1$ equations in the $n+1$ unknowns $a_0, \ldots, a_n$. Since the vectors $\{(1, r_{i,1}, \ldots, r_{i,n}) \in (\mathbb{F}_q)^{n+1} : i = 1, \ldots, n+1\}$ are linearly independent, it follows that $G - G'$ is the zero polynomial, so $G = G'$. Hence $F = F'$, as desired. □

A.1. Basic Construction: Proof of Linear Independence. We use assumptions and notation as in the proof of Theorem 2.11. Recall that the information obtained by the coalition C is contained in equation sets (1) and (3), together with, for each $U_h \notin C$, equation set (5). We have a total of $n\omega\psi + n\omega + \omega + \psi n$ equations, which would imply there are at least $n - \omega$ free variables in the given linear system.

We proceed by showing that allowing C access to an additional $n - \omega$ equations (in the form of sample signatures from each user not in C) suffices to solve the linear system. This implies the linear independence of the original set of equations, as desired.

Lemma A.3. *Let $U_h \notin C$. Suppose C has access to an additional h-authentic signature from U_h on some message $m_{h,\psi+1}$ satisfying $m_{h,\psi+1} \neq m_{h,k}$ for $1 \leq k \leq \psi$. Then this is equivalent to C having access to all of the signing information from U_h.*

Proof. This follows immediately from the fact that U_h's signing algorithm $s_h(\vec{y}, z)$ is a polynomial of degree $\psi + 1$ in z. □

Lemma A.3 implies that the system of equations

$$\{C_{ikh} : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 1 \leq h \leq \omega\} \cup \{D_{0kh} : 0 \leq k \leq \psi, 1 \leq h \leq n\}$$

is equivalent to the original system of equations known to C, plus $n - \omega$ additional equations $\left\{B_{0h(\psi+1)} : U_h \notin C\right\}$ (obtained from an extra h-authentic signature on some new message $m_{h,\psi+1}$ for each $U_h \notin C$). In the following lemma, we show this new set is linearly independent, and therefore the linear independence of the original set follows.

Lemma A.4. *The coefficient matrix formed from the equations*

$$\{C_{ikh} : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 1 \leq h \leq \omega\} \cup \{D_{0kh} : 0 \leq k \leq \psi, 1 \leq h \leq n\}$$

has nonzero determinant.

Proof. The coefficient matrix E is a block matrix of the form

$$E = \left[\begin{array}{c:c} A & 0 \\ \hdashline C & D \end{array}\right],$$

where A and D are square matrices. Thus the determinant of the coefficient matrix, $\det(E)$, is defined by $\det(E) = \det(A)\det(D)$. We show that $\det(E) \neq 0$.

Here the submatrix

$$\left[\, A \,\vdots\, 0 \,\right]$$

is derived from the equations $\{D_{0kh} : 0 \leq k \leq \psi, 1 \leq h \leq n\}$, where

$$A = \begin{bmatrix} V_n & 0 & \cdots & 0 \\ 0 & V_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_n \end{bmatrix}$$

is a diagonal matrix with $(\psi + 1)$ Vandermonde matrices V_n on the diagonal. That is, we have

$$V_n = \begin{bmatrix} 1 & U_1 & \cdots & U_1^{n-1} \\ 1 & U_2 & \cdots & U_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & U_n & \cdots & U_n^{n-1} \end{bmatrix}.$$

To see that A is invertible, note that $\det(A) = \prod_{i=1}^{(\psi+1)} \det(V_n) \neq 0$.

The submatrix

$$\left[\, C \,\vdots\, D \,\right]$$

is derived from the equations $\{C_{ikh} : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 1 \leq h \leq \omega\}$. The matrix D is defined by

$$D = \begin{bmatrix} v_{1,1}\,\mathrm{I} & v_{1,2}\,\mathrm{I} & \cdots & v_{1,\omega}\,\mathrm{I} \\ v_{2,1}\,\mathrm{I} & v_{2,2}\,\mathrm{I} & \cdots & v_{2,\omega}\,\mathrm{I} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\omega,1}\,\mathrm{I} & v_{\omega,2}\,\mathrm{I} & \cdots & v_{\omega,\omega}\,\mathrm{I} \end{bmatrix},$$

where I is the $n(\psi + 1) \times n(\psi + 1)$ identity matrix.

The fact that $\det(D) \neq 0$ follows immediately from the the linear independence of the coalition's verification keys, $\{\vec{v_h} : 1 \leq h \leq \omega\}$. □

Remark A.1. The security analysis for a general coalition (whose verification keys may or may not be linearly independent) is very similar. Recall the assumption that the n elements $\vec{v_1}, \ldots \vec{v_n} \in (\mathbb{F}_q)^\omega$ satisfy the additional property that for any subset of size $\omega + 1$, the corresponding subset of size $\omega + 1$ formed from the new vectors $[1, \vec{v_1}], \ldots, [1, \vec{v_n}] \in (\mathbb{F}_q)^{\omega+1}$ is a linearly independent set. Consider a possible coalition C of size ω, where $V = \{\vec{v_h} : U_h \in C\}$ is the set of C's verification keys. Then Lemma A.1 implies that, for any $\vec{v_r} \notin V$, we can pick a subset of size ω with full rank from $V \cup \{\vec{v_r}\}$.

There are then two cases. Either the set V has rank ω, so that V forms a basis for $(\mathbb{F}_q)^\omega$, or the additional vector $\vec{v_r}$ is needed to form a basis. In the former case, the analysis is as above. In the latter, the span of V is a subspace of $(\mathbb{F}_q)^\omega$ of dimension $\omega - 1$. The linear system corresponding to C's information has $n - (\omega - 1)$ free variables, which can be shown using a linear algebra trick similar to the one used above.

In fact, if we did not have any additional assumptions on user verification keys (other than that they are chosen uniformly at random from $(\mathbb{F}_q)^\omega$), the proof follows much as before. A coalition C's information in this case depends on the rank of V, i.e., the linear system has $n - r$ free variables, where $r = \text{rank}(V)$.

A.2. Key Insulation Construction: Proof of Linear Independence. We use assumptions and notation as in the proof of Theorem 2.12. Recall that the information obtained by the coalition C is contained in the following sets of equations: sets (6) and (8), together with, for each $U_h \notin C$, one of set (9) or set (10) (depending on the type of key exposure), and set (11). We have a total of $n\omega(\psi+1)(\gamma+1) + \omega(\psi+1)(\gamma+1) + (n-\omega)\gamma(\psi+1) + (n-\omega)\psi$ equations, which implies that we have $n - \omega$ free variables in the given linear system.

We use the same method as in Section A.1; we include the argument here for completeness. We proceed by showing that allowing C access to an additional $n - \omega$ equations (in the form of sample signatures from each each user not in C) suffices to solve the linear system. This implies the linear independence of the original set of equations, as desired.

Lemma A.5. *Let $U_h \notin C$. Suppose C has access to an additional h-authentic signature from U_h on some message $m_{h,\psi+1}$ satisfying $m_{h,\psi+1} \neq m_{h,k}$ for $1 \leq k \leq \psi$ in addition to either master key or signing key exposure from U_h. Then this is equivalent to C having access to all of the signing information from U_h.*

Proof. Consider a user $U_h \notin C$. Then C has access to up to ψ sample signatures from U_h on distinct messages $m_{h,k}$ for $1 \leq k \leq \psi$, which yield the equations $\{B_{0hk'} : 1 \leq k' \leq \psi\}$. Suppose C has access to one additional signature from U_h, yielding the additional equation $B_{0h(\psi+1)}$.

Now suppose C has achieved master key exposure for U_h. Then the coalition C has access to the set $\{D_{0k\ell h} : 0 \leq k \leq \psi, 1 \leq \ell \leq \gamma\}$. Let

$$B_h = \{D_{0k\ell h} : 0 \leq k \leq \psi, 1 \leq \ell \leq \gamma\} \cup \{B_{0hk'} : 1 \leq k' \leq \psi\}.$$

We show that the coalition having access to the set $B_h \cup \{B_{0h(\psi+1)}\}$ is equivalent to C knowing all of the signing information for U_h, namely the set $\{D_{0k\ell h} : 0 \leq k \leq \psi, 0 \leq \ell \leq \gamma\}$. First note that these two sets are both of cardinality $(\psi+1)(\gamma+1)$.

It is easy to see that the equations in $B_h \cup \{B_{0h(\psi+1)}\}$ may be written as a linear combination of the equations in $\{D_{0k\ell h} : 0 \leq k \leq \psi, 0 \leq \ell \leq \gamma\}$. To see that $B_h \cup \{B_{0h(\psi+1)}\}$ suffices to derive the signing information of U_h, note that there are $\psi+1$ equations $\{B_{0hk'} : 1 \leq k' \leq \psi+1\}$ in the $\psi+1$ unknowns $\{D_{0k0h} : 0 \leq k \leq \psi\}$. (The linear independence of these equations is guaranteed so long as the messages chosen for the sample signatures from U_h are distinct.)

Now suppose C has signing key exposure for U_h instead of master key exposure. Let

$$B'_h = \{E_{0kt_{h_d}} : 0 \leq k \leq \psi, 1 \leq d \leq \gamma\} \cup \{B_{0hk'} : 1 \leq k' \leq \psi\}.$$

It is easy to see that the equations in $B_h \cup \{B_{0h(\psi+1)}\}$ may be written as a linear combination of the equations in $\{D_{0k\ell h} : 0 \le k \le \psi, 0 \le \ell \le \gamma\}$ and that these two sets have the same cardinality. To see that $B_h \cup \{B_{0h(\psi+1)}\}$ suffices to derive the signing information of U_h, note that there are $\psi + 1$ equations $\{B_{0hk'} : 1 \le k' \le \psi + 1\}$ and $\gamma(\psi + 1)$ equations $\left\{E_{0kt_{h_d}} : 0 \le k \le \psi, 1 \le d \le \gamma\right\}$ in the $(\psi + 1)(\gamma + 1)$ unknowns $\{D_{0k\ell h} : 0 \le k \le \psi, 0 \le \ell \le \gamma\}$. (The linear independence of these equations is guaranteed so long as the messages chosen for the sample signatures from U_h are distinct.) □

The following lemma completes the result:

Lemma A.6. *The coefficient matrix formed from the equations*

$$\{C_{ik\ell h} : 0 \le i \le n-1, 0 \le k \le \psi, 0 \le \ell \le \gamma, 1 \le h \le \omega\}$$
$$\cup \{D_{0k\ell h} : 0 \le k \le \psi, 0 \le \ell \le \gamma, 1 \le h \le n\}$$

has nonzero determinant.

Proof. The coefficient matrix E is a block matrix of the form

$$E = \left[\begin{array}{c:c} A & 0 \\ \hdashline C & D \end{array}\right],$$

where A and D are square matrices. Thus the determinant of the coefficient matrix, $\det(E)$, is defined by $\det(E) = \det(A)\det(D)$. We show that $\det(E) \neq 0$.

Here the submatrix

$$\left[\begin{array}{c:c} A & 0 \end{array}\right]$$

is derived from the equations $\{D_{0k\ell h} : 0 \le k \le \psi, 0 \le \ell \le \gamma, 1 \le h \le n\}$, where

$$A = \begin{bmatrix} V_n & 0 & \cdots & 0 \\ 0 & V_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_n \end{bmatrix}$$

is a diagonal matrix with $(\psi + 1)(\gamma + 1)$ Vandermonde matrices V_n on the diagonal. That is, we have

$$V_n = \begin{bmatrix} 1 & U_1 & \cdots & U_1^{n-1} \\ 1 & U_2 & \cdots & U_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & U_n & \cdots & U_n^{n-1} \end{bmatrix}.$$

To see that A is invertible, note that $\det(A) = \prod_{i=1}^{(\psi+1)(\gamma+1)} \det(V_n) \neq 0$.

The submatrix

$$\left[\begin{array}{c:c} C & D \end{array}\right]$$

is derived from the equations $\{C_{ik\ell h} : 0 \leq i \leq n-1, 0 \leq k \leq \psi, 0 \leq \ell \leq \gamma, 1 \leq h \leq \omega\}$. The matrix D is defined by

$$D = \begin{bmatrix} v_{1,1}\,\mathrm{I} & v_{1,2}\,\mathrm{I} & \cdots & v_{1,\omega}\,\mathrm{I} \\ v_{2,1}\,\mathrm{I} & v_{2,2}\,\mathrm{I} & \cdots & v_{2,\omega}\,\mathrm{I} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\omega,1}\,\mathrm{I} & v_{\omega,2}\,\mathrm{I} & \cdots & v_{\omega,\omega}\,\mathrm{I} \end{bmatrix},$$

where I is the $n(\psi+1)(\gamma+1) \times n(\psi+1)(\gamma+1)$ identity matrix.

The fact that $\det(D) \neq 0$ follows immediately from the the linear independence of the coalition's verification keys, $\{\vec{v_h} : 1 \leq h \leq \omega\}$. □