The Design and Analysis of Hash Families

For Use in Broadcast Encryption

by

Devon James O'Brien

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Masters of Science in Computer Science

Approved May 2012 by the
Graduate Supervisory Committee:

Charles Colbourn, Chair
Rida Bazzi
Andrea Richa

ARIZONA STATE UNIVERSITY

May 2012

ABSTRACT

Broadcast Encryption is the task of cryptographically securing communication in a broadcast environment so that only a dynamically specified subset of subscribers, called the privileged subset, may decrypt the communication. In practical applications, it is desirable for a Broadcast Encryption Scheme (BES) to demonstrate resilience against attacks by colluding, unprivileged subscribers. Minimal Perfect Hash Families ($PHFs$) have been shown to provide a basis for the construction of memory-efficient t-resilient Key Pre-distribution Schemes ($KPSs$) from multiple instances of 1-resilient $KPSs$. Using this technique, the task of constructing a large t-resilient BES is reduced to finding a near-minimal PHF of appropriate parameters. While combinatorial and probabilistic constructions exist for minimal $PHFs$ with certain parameters, the complexity of constructing them in general is currently unknown.

This thesis introduces a new type of hash family, called a Scattering Hash Family ($ScHF$), which is designed to allow for the scalable and ingredient-independent design of memory-efficient $BESs$ for large parameters, specifically resilience and total number of subscribers. A general BES construction using $ScHFs$ is shown, which constructs t-resilient $KPSs$ from other $KPSs$ of any resilience $1 \leq w \leq t$.

In addition to demonstrating how $ScHFs$ can be used to produce $BESs$, this thesis explores several $ScHF$ construction techniques. The initial technique demonstrates a probabilistic, non-constructive proof of existence for $ScHFs$. This construction is then derandomized into a direct, polynomial time construction of near-minimal $ScHFs$ using the method of conditional expectations. As an alternative approach to direct construction, representing $ScHFs$ as a k-restriction

problem allows for the indirect construction of $ScHFs$ via randomized post-optimization.

Using the methods defined, $ScHFs$ are constructed and the parameters' effects on solution size are analyzed. For large strengths, constructive techniques lose significant performance, and as such, asymptotic analysis is performed using the non-constructive existential results. This work concludes with an analysis of the benefits and disadvantages of $BESs$ based on the constructed $ScHFs$. Due to the novel nature of $ScHFs$, the results of this analysis are used as the foundation for an empirical comparison between $ScHF$-based and PHF-based $BESs$. The primary bases of comparison are construction efficiency, key material requirements, and message transmission overhead.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

GLOSSARY OF TERMS

BES	–	Broadcast Encryption Scheme
KPS	–	Key Pre-distribution Scheme
TA	–	Trusted Authority
PHF	–	Perfect Hash Family
$ScHF$	–	Scattering Hash Family
XOR	–	Exclusive-Or
TTS	–	Traitor-Tracing Scheme
CA	–	Covering Array
SHF	–	Separating Hash Family
DHF	–	Distributing Hash Family
BEO	–	Broadcast Encryption Overhead
SIO	–	Set Identification Overhead
KMO	–	Key Material Overhead
KpU	–	Keys Per User

Chapter 1

INTRODUCTION

A broadcast environment is an overarching title given to any scenario in which a central authority is attempting to communicate over a channel that cannot be guaranteed to be private. Such an environment can be physical, as with the distribution of DVDs or other physical media, or it can be virtual, as with radio and network-based broadcasts. In many of these scenarios, it is desirable to limit the ability of parties to receive a given message to a select subset of listeners while still taking advantage of the convenience of transmitting over a broadcast medium. Traditional symmetric encryption models do not fit this model of communication because they tend to operate by securing individual channels of communication between parties. This approach would result in an infeasible amount of overhead for the central authority to send a message to potentially tens of thousands of listeners. In pursuit of this goal, and in light of these restrictions, the study of broadcast encryption was formed.

When broadcasting a message that is to be encrypted to some set of listeners, there are two important factors to consider. The first is how to distribute the key material to the listeners and the second is how the central authority encrypts and transmits this message across the environment. As such, a Broadcast Encryption Scheme (BES) is defined in two phases: the key pre-distribution phase and the broadcast phase. When the central authority is acting as the broadcaster under these schemes, it is called the Trusted Authority (TA) since it is then responsible for managing key material and encryption for its listeners. These listeners are then referred to as subscribers based on their need for initial interaction with the TA. Trivial solutions exist for broadcast encryption;

however, in general, optimizing key pre-distribution comes at a cost of performance in the broadcast phase and vice versa. The goal for BES design is to arrive at a secure and efficient trade-off between these phases. Additionally, more sophisticated designs offer a security assurance to the design called resilience. Resilience, or when a specific quantity is expressed, t-resilience is a measure of the number of misbehaving subscribers that are required to compromise the security of a BES, typically by sharing their key material with one another.

The majority of research into broadcast encryption schemes has gone into the areas of designing 1-resilient schemes, designing standalone t-resilient schemes, and designing t-resilient schemes that use 1-resilient schemes as a base ingredient. In the 1990s, a strong relationship between $BESs$ and the existence of certain combinatorial structures called hash families was established [1], and subsequently, much research has gone into strengthening the knowledge of these two fields. Specifically, Perfect Hash Families ($PHFs$) have been demonstrated to provide the basis for a BES construction that is efficient in both broadcast and key material overhead. The separation property of a PHF allows for the construction of a k-resilient BES from smaller instances of 1-resilient schemes. Designing efficient algorithms for the generation of minimal or near-minimal $PHFs$ has been a well-studied problem in the last several decades. However, much is still unknown about their construction in general.

While $PHFs$ have been shown to be very efficient in constructing t-resilient schemes from 1-resilient schemes, no analogous combinatorial structure has been created that allows for the inflation of a w-resilient scheme into a similarly efficient t-resilient scheme for $w > 1$. Such a structure would generalize $PHFs$,

but does not appear to be provided by the body of existing PHF generalizations. This thesis presents a new type of hash family, called a Scattering Hash Famiy ($ScHF$) that is shown to exhibit the desired inflation behavior. In addition to providing an existential analysis for these families using a variety of techniques, this thesis provides a direct comparison between $ScHF$-based and PHF-based $BESs$ using both the broadcast and key material overheads as metrics of performance.

Overview of this Thesis

Chapter 2 provides the basics for framing the study of hash family based broadcast encryption. Perfect hashing is introduced to provide a foundation for both the canonical PHF-based BES as well as for the design and subsequent analysis of Scattering Hashing.

Chapter 3 formalizes Broadcast Encryption Schemes and establishes the relationship between these schemes and $PHFs$. From here, this relationship is generalized and subsequently, Scattering Hash Families are introduced, formalized, and analyzed for their broadcast encryption properties.

Chapter 4 details the existential conditions and construction of $ScHFs$. The initial constructive and probabilistic proofs of existence for this type of hash family are provided. A randomized construction algorithm for $ScHFs$ is provided and subsequently, using the technique of derandomization, this algorithm is used to create an efficient deterministic algorithm for constructing $ScHFs$.

Chapter 5 incorporates the construction techniques as a foundation for the first empirical existential analysis for $ScHFs$. Subsequently, these results are utilized in constructing a comparison between PHF-based and $ScHF$-based $BESs$

on the metrics of Key Material Overhead, Broadcast Encryption Overhead, and Information Rates.

Chapter 6 concludes this thesis with a discussion of the results and a brief discussion of future work.

Chapter 2

BACKGROUND

2.1 Perfect Hashing

Originally motivated by optimization of compiler design, the study and design of Perfect Hash Families ($PHFs$) have since been extended to many different applications ranging from combinatorial design to cryptography. In addition to expanding their uses, much research has gone into generalized constructions as well as bounds on the various parameters [2] [3]. Of particular relevance to this work is the construction of minimal and near-minimal $PHFs$ and their applications to the study of Broadcast Encryption. The relevance of perfect hashing to broadcast encryption was established by Fiat and Naor [1] in 1994 and has stood as one of the predominant foundations for generating t-resilient $BESs$ provided a minimal or near-minimal PHF is known for the given parameters.

Formally, a $(N;\ k, v, t) - Perfect\ Hash\ Family$ is a set of functions F such that $\forall f \in F$:

$$f: \{1, \dots, k\} \to \{1, \dots, v\}$$

and for any subset $C \subseteq \{1, \dots, k\}$ with $|C| = t$, $\exists f \in F$ such that f is an injection on C. While a perfect hash function is one that maps every element of its domain to a unique element of its range, Perfect Hash Families can be viewed as a relaxation of this requirement. By necessity, a perfect hash function $f: \{1, \dots, k\} \to \{1, \dots, v\}$ would at minimum require $v \geq k$, but in most cases, constructions produce $v \gg k$, which is prohibitively restrictive in application. Not only are such functions often difficult to define on a large universe of inputs, but they also require a large amount of memory when $v \gg k$. By relaxing this property to allow multiple functions with the property that at least one such function will be injective

for any set of t elements of the domain, this overhead is greatly reduced. By convention [3], all Perfect Hash Families in this work are denoted $PHF(N; k, v, t)$, which in the absence of a universally accepted representation is the most common form.

At times, it is more convenient to represent a PHF as an array of elements subject to a separation constraint for all subsets of elements of a certain size. When viewed as an array, a $(N; k, v, t) - Perfect\ Hash\ Family$ is an $N \times k$ array populated with symbols from V where $|V| = v$ in such a way that for any selection of t columns of the matrix, there exists at least one row such that the symbols contained in the intersection of this row and the selected columns are all distinct. Both the injective property of the function definition and the requirement for distinct elements in a row for the t-subsets are different ways of stating the PHF separation condition. Intuitively, the separation condition is what distinguishes a PHF from all other varieties of hash families. In Section 3.2, the PHF separation condition is generalized in an as-yet unexplored fashion in the construction of $ScHFs$.

Many techniques have been developed to generate $PHFs$, ranging from combinatorial construction [4], probabilistic construction [5], to direct algorithmic approaches [6]. Each of these techniques suffers from unique drawbacks, which prevents their sole use in generalized PHF construction. Combinatorial constructions tend to produce elegant, simple, and often minimal instances of $PHFs$, however, they are highly restrictive on the relationships of the parameters of the PHF and as such, do not generalize well. Figure 2.1.1 below depicts a minimal $PHF(8; 6, 4, 4)$ that can be generated by such a method. Probabilistic construction is a general term for two different probabilistic approaches; the first

of which offers a probabilistic guarantee on the separation for all t-subsets [5], while the second guarantees separation while probabilistically assuring minimality [6]. The latter of these two approaches has produced the best known general bound for $PHFs$ and in Sections 4.2 and 5.1, it is the most successful technique employed for $ScHF$ construction in this thesis.

1	**2**	**3**	**4**	**5**	**6**
1	1	1	2	3	1
1	2	3	2	4	1
1	2	3	3	2	3
1	1	1	2	3	1
3	3	3	4	2	4
3	4	2	1	2	1
4	4	4	2	4	4
4	3	1	4	3	3

Table 2.1.1: A Minimal $PHF(8;6,4,4)$

For the applications considered in this work, probabilistic guarantees of separation for a PHF violate the provable perfect secrecy of $BESs$ utilizing these $PHFs$ to determine key pre-distribution. Moreover, these structures are not even guaranteed to be $PHFs$ because of this property; however, some applications can handle this weakening by accepting the risk that certain small subsets of unprivileged users can decrypt the content [7]. $PHFs$ that separate all $\binom{k}{t}$ t-subsets but only probabilistically assure minimality are often the result of greedy or derandomized constructions to efficiently generate $PHFs$. For PHF-based $BESs$, this allows for the possibility that users are forced to store significantly more key material than is necessary for the scheme being deployed. These

properties are examined in greater detail in Section 3.1 and Section 5.3 respectively.

2.2 Broadcast Encryption

Broadcast Encryption is the cryptographic problem wherein a centralized Trusted Authority (TA) desires to transmit a message across a broadcast medium that is encrypted in such a fashion that only a particular, dynamic subset of subscribed listeners can decrypt and observe the message. Such a scheme not only needs to protect against non-subscribed listeners, but also against valid, registered subscribers who are not entitled to decrypt the contents of a given message. Formally, a Broadcast Encryption Scheme is represented as $(k, t) - BES$, indicating it is a scheme on k subscribed listeners with a resilience against colluding parties of size at most t. Traditionally, the broadcast message is the encryption key to a large message that has been encrypted with a strong symmetric algorithm such as AES [8], which is broadcast after the secure distribution of the encryption key. For this reason, it is often the goal to restrict the focus of designing $BESs$ to those in which a single message M is chosen to persist throughout a large amount of content distribution with infrequent modifications to the privileged subset. The most famous instance of this type of scheme is the AACS content protection scheme applied to Blu-Ray discs [8]. In this scheme, a sufficiently large BES size is chosen and the KPS is deployed to each licensed Blu-Ray player manufacturer. Each Blu-Ray disc is encrypted with a key M and the sale combined with the ease of copying the encrypted content on the discs is analogous to a broadcast in the traditional sense.

While the size of a BES is determined simply by the number of subscribers, resilience is crucial, but is not so easily determined. In [9], Luby and Staddon prove lower bounds on key material requirements and message overhead in the circumstances of $k \gg t$ or $k - t \ll k$, which are reasonable bounds for pay-per-view TV $BESs$, but not necessarily for all applications. Although it is true for most memory efficient schemes that selecting a higher resilience results in the need for much higher amounts of pre-distributed key material, in practice, the selection of resilience for a BES is rooted deeper in procedural, practical, or economic restrictions than it is in mathematics [9] [10]. Consider a pay-per-view TV service's broadcast encryption model. Subscribers are customers of the content provider (TA) who have registered for this service and have had a box delivered to their house, which among other functions, serves as a tamper-proof storage device for the subscriber's key material. The necessary resilience in this situation is based on a risk analysis of subscribers successfully tampering with their boxes, spoofing registration to obtain multiple boxes, and reaching out undetected to other parties desiring to circumvent the scheme. If these factors can be mitigated to a nominal degree, the BES deployed can utilize a smaller resilience.

An analysis of broadcast encryption would be incomplete without considering the varied extensions of broadcast encryption that have been discovered since its inception. In its initial form, broadcast encryption was based solely around the concept of providing resilience against a colluding party of unprivileged users of at most a certain size [1]. When the colluding party exceeds this threshold, this subset of unprivileged users is able to freely decrypt content at will. Traitor-tracing [11] [12] [13] is a natural extension to resilience, and allows

the TA to identify some subset of the colluding party when a compromise occurs and prevents these members from framing an innocent subscriber for their actions. This technique is widely used in protecting against unlawful reproduction of licensed software [14]. These schemes put members of a colluding party in direct risk of discovery, which effectively protects against unwanted distribution by severely de-incentivizing this behavior. Stinson, Trung, and Wei [15] provide a detailed analysis of the use of hash families in the production of frame-proof and traitor-tracing codes, a key ingredient in the construction of several such Traitor-Tracing Schemes ($TTSs$).

In addition to the ability to identify adversarial subscribers, it is desired that methods of broadcast encryption include the ability to revoke a set of keys associated with one or more subscribers. Revocation is the ability to remove a subscriber's ability to decrypt all future broadcasts by rendering those keys useless. In simple $BESs$, this can be performed at the TA by removing any revoked subscribers from the privileged subset P before broadcasting the message. Simple $BESs$, however, lack the ability to actually trace a traitor, since the fully decrypted content is the same for all users. The AACS [8] BES previously mentioned incorporates both of these concepts into an efficient trace-revoke scheme that can not only detect the type of Blu-Ray player that has been compromised, but will also render the class of Blu-Ray players used in this compromise unable to play any future releases.

Formally, a $(k,t) - BES$ is defined to be a broadcast encryption scheme with k pre-registered subscribers that must be resilient against colluding parties of non-privileged listeners of size at most t. A $(k,t) - BES$ consists of two phases, the first of which is the pre-distribution phase during which keys are

generated, arranged, and distributed to the set of subscribers K. The second phase is the broadcast phase in which a message M is produced and encrypted based on the desired privileged subset $P \subseteq K$ and is then transmitted across the broadcast medium. While, in general, BES resilience can be defined as a probabilistic guarantee that colluding parties cannot decrypt a particular message [7], the scope of this thesis restricts this definition to deterministic resilience so that, definitively, no t-subset of unprivileged subscribers is able to decrypt any broadcast message M. There exist several trivial solutions to this problem [1] [16], two of which are provided below. Each of these schemes represents one extreme in the trade-off of key material overhead versus broadcast length. In practice, both of these extremes are avoided in favor of schemes that provide an efficient compromise between these two factors.

Pre-distribution Phase:

- Let K be the set of all subscribers of the BES.
- Generate K distinct symmetric keys uniformly at random and pre-distribute one to each subscriber $s \in K$.

Broadcast Phase:

- For broadcast message M, let $P \subseteq K$ be the privileged subset.
- When sending message M, the TA produces $|P|$ copies of the message and encrypts each copy with the key of a member of the privileged subset.
- The TA broadcasts all $|P|$ encrypted versions of M in succession: $E_1(M), E_2(M), \dots, E_{|P|}(M)$.

Figure 2.2.1: A Naïve BES with Large Broadcast Encryption Overhead

Pre-distribution Phase:

- Let K be the set of all subscribers of the BES.
- Generate $2^{|K|}$ distinct symmetric keys uniformly at random, each corresponding to one possible subset of subscribers from the set of all subsets of K, $\mathcal{P}(K)$.
- For every subscriber $s \in K$, let $\mathcal{P}_s(K)$ denote the set of subsets that contain subscriber s. For all s, distribute to this subscriber every symmetric key corresponding to a subset in $\mathcal{P}_s(K)$.

Broadcast Phase:

- For broadcast message M, let P be the privileged subset of subscribers.
- When sending message M, the TA selects the key corresponding to the privileged subset $P \in \mathcal{P}(K)$ and encrypts M with this key
- The TA broadcasts the single message $E_P(M)$.

Figure 2.2.2: A Naïve BES with Large Key Material Overhead

The first method with large broadcast overhead is perhaps the most intuitive BES design. Each subscriber gets one personal symmetric key that the TA uses to encrypt a copy of the message M. In this scheme, the broadcast message must be encrypted and re-broadcast for each user in the privileged subset P. This scheme's broadcast overhead thus scales linearly with both the size of the broadcast message as well as the size of P, which is prohibitively expensive. The second method with large key material overhead optimizes the

broadcast down to a single message; however, even for small instances of this scheme, each subscriber's key storage is $2^{|K|-1}$ keys, which is, once again, prohibitively expensive. Fiat and Naor [1], among others, demonstrated that for broadcast encryption to be practical, an efficient trade-off between these two factors must be obtained.

2.3 Randomization and Derandomization

Randomization is a powerful tool in the design of algorithms that, rather than relying on making decisions on the input in a fixed, iterative fashion, instead harnesses a secondary input of uniformly distributed random bits to govern the operation or output of the algorithm. Randomized algorithms are split into two major categories based on how the randomness is utilized: Monte Carlo algorithms offer deterministic run time while only probabilistically guaranteeing veracity of output whereas Las Vegas algorithms may fail to terminate but always produce a correct result upon termination. Historically, the primary motivating factor in designing randomized algorithms is the desire to obtain practical results in an efficient manner in the absence of a known efficient deterministic technique. Contextual evidence of this phenomenon exists in the study of primality testing of integers [17] [18], escaping local minima/maxima during Simulated Annealing [19], and in the construction of convex hulls and polytopes, all of which produced efficient randomized algorithms years or even decades before deterministic techniques of equivalent computational complexity were discovered.

The probabilistic guarantees provided by these algorithms are sometimes undesirable in practice. Derandomization involves removing the randomness from a probabilistic algorithm while maintaining or even improving upon its

performance in either solution strength or computational complexity. Of particular importance to this work is the derandomization of probabilistic, non-constructive proofs of existence into efficient and constructive deterministic algorithms. Probabilistic non-constructive proofs of existence for various hash families have been analyzed in depth [5], and in many cases yield the lowest known bounds for minimality. While these results are useful for asymptotic analysis, applications of these hash families require actual constructions to be utilized. Despite this fact, the analysis of randomized algorithms may reveal core properties inherent to a particular combinatorial problem. In certain cases, these properties can be harnessed in a deterministic fashion as long as certain constraints are maintained during the derandomization.

Section 4.1 describes a randomized proof of existence for $ScHFs$ that follows a general approach shown to be successful for many other types of hash families [6]. This proof analyzes the probability that a randomly generated array will fail to meet the $ScHF$ separation condition for all t-subsets of columns. As the number of rows increases, this probability decreases and once it reaches a certain point, there must exist some array of this size that meets the separation condition and thus, a $ScHF$ of the current parameters must exist. By utilizing the Method of Conditional Expectation [20], this proof is systematically derandomized into a polynomial-time deterministic $ScHF$ construction algorithm, which is subsequently analyzed in Section 5.1.

2.4 Post-Optimization and k-Restrictions

In combinatorial design, it is often the case that when analyzing a new structure, the existing construction methods produce sub-optimal results. Optimality in

terms of hash family construction is measured in output size. The smaller the constructed family, the stronger it is considered. Advanced techniques for constructing small instances tend to evolve out of earlier naïve approaches [6]. For hash families as well as other array-based constraint satisfaction designs, constructions tend to suffer from a trade-off between simplicity of construction and runtime, guarantee of constraint satisfaction, and minimality [2] [5] [6]. When constructing instances of these designs on large parameters, the time complexity of the chosen algorithm becomes increasingly important. As demonstrated in the derandomized $ScHF$ and PHF construction algorithms in Section 4.2, ensuring that execution occurs in an efficient manner tends to sacrifice minimality of design in favor of polynomial time complexity.

Despite the fact that the constructions produced by these algorithms are in general not minimal, these results can be refined via a technique called post-optimization. Post-optimization is a type of combinatorial optimization approach that operates *a posteriori* on the output of a separate construction for a given combinatorial design. Essentially, it is the technique of taking a suboptimal solution to open problems such as covering array or hash family construction and improving upon the strength of the solution.

Post-optimization has been shown to be highly successful for improving known bounds of minimality for Covering Arrays (CAs) and several well-known forms of hash families including Perfect, Separating, and Distributing Hash Families; referred to as $PHFs$, $SHFs$, and $DHFs$ respectively [21]. This work defines a randomized post-optimization technique that operates on a set of designs called k-restriction problems. Each of the structures are shown to be instances of a k-restriction problem, and once represented as such, are post-

optimized using necessity analysis on a symbol-by-symbol basis. The optimization on the array occurs when an entire row is determined to be unnecessary to the structure and is discarded, thus providing a smaller instance, which is a stronger solution. Formally, k-restriction problems are defined as follows [22]:

1. The input is an alphabet Σ of size $|\Sigma| = \mathrm{q}$, a length m, and a set of s possible demands: $f_i: \Sigma^{\mathrm{k}} \to \{0,1\}$, $1 \leq i \leq s$. For every $1 \leq i \leq s$, there exists $a \in \Sigma^{\mathrm{k}}$ so that $f_i(a) = 1$.
2. The task is to prepare a set $A \subseteq \Sigma^{\mathrm{m}}$ so that: For any choice of k indices $1 \leq i_i < \cdots < i_k \leq m$, and a demand j, $1 \leq j \leq s$, there is some $a \in A$ such that $f_j\big(a(i_1) \ldots a(i_k)\big) = 1$.
3. The smaller $|A|$ is, the higher the quality of the solution.

Figure 2.4.1: Definition of k-Restriction Problems

By formalizing Scattering Hash Families as a k-restriction problem and subsequently performing k-restriction-based post-optimization on known instances, the non-minimal instances generated by the derandomized construction in Section 4.2 are post-optimized in an attempt to strengthen the known bounds of minimality for $ScHFs$. The results of this post-optimization are analyzed in Section 5.2.

Chapter 3

BROADCAST ENCRYPTION

3.1 Fiat-Naor Broadcast Encryption

The Fiat-Naor BES [1] is essentially a technique of extending a 1-resilient KPS to a t-reslilient BES through the use of Perfect Hash Families. In order to accomplish this, an arbitrary 1-reslient KPS is selected as the base ingredient for the larger scheme. From here, the construction requires a PHF whose parameters match those of the desired BES; that is whose number of columns equals the number of subscribers and whose strength corresponds to the desired resilience of the BES. The innovative aspect of this technique is revealed in how the provided PHF is used to inflate the resiliency. The rows of the PHF are treated as partitions of the columns based on the symbols appearing in the PHF. From here, every partition of every row is assigned an independent 1-resilient KPS. In their initial work [1], Fiat and Naor demonstrate several such ingredient $KPSs$. Subsequently, these schemes have been analyzed in depth [16] and other memory-efficient alternatives have been proposed [23].

The result of applying the ingredient $KPSs$ is that each user is assigned keys for every partition of the PHF in which they are present. In order to securely broadcast a message to an arbitrary privileged subset of users, the TA randomly generates components $M_1, \dots, M_N$ so that $\bigoplus_{i=1}^{N} M_i = M$ where N is the number of rows of the PHF. The components of the message are then encrypted with the keys in such a fashion as to allow only privileged users the ability to decrypt, while offering no information about M to colluding parties of size t or smaller. A detailed proof of this concept is given in Section 3.5 in the context of $ScHF\ BESs$, however, the information-theoretic properties are the same.

In order to analyze the Fiat-Naor construction as well as to provide a concrete ingredient KPS for the $ScHF\ BES$ construction in Section 3.3, the following process details the key pre-distribution and broadcast protocol of a $PHF(3; 5, 2, 2)\ BES$. In a broadcast environment consisting of $k = 5$ subscribers, it is desired to create a BES whose broadcasts are resilient against colluding parties of unprivileged subscribers of size $t = 2$ or smaller. The KPS will be constructed from the PHF in Figure 3.1.1.

1	**2**	**3**	**4**	**5**
1	1	1	2	2
1	1	2	1	2
1	2	1	1	2

Table 3.1.1: A $PHF(3;\ 5, 2, 2)$

Once the PHF has been obtained, for each row R_i, $1 \leq i \leq N$, partition the column indices based on the elements appearing in this row. Let p_{ij} be the label for the j^{th} partition in row R_i. On each partition p_{ij}, create an instance of a 1-resilient KPS on k_{ij} subscribers where $k_{ij} = |p_{ij}|$ is the size of the partition and distribute the keys according to this KPS.

$p_{11} = \{1, 2, 3\}$	$\{4, 5\} = p_{12}$
$p_{21} = \{1, 2, 4\}$	$\{3, 5\} = p_{22}$
$p_{31} = \{1, 3, 4\}$	$\{2, 5\} = p_{32}$

Table 3.1.2: Symbolic Partitions for $PHF(3; 5, 2, 2)$

Let the 1-resilient ingredient KPS selected be the following: for each member of partition p_{ij}, first distribute a Null-key K_{ij0} to each member. The Null-key prevents subscribers from outside of this partition from recovering the message component M_i to be broadcast on this partition. Then, for each subscriber s present in the partition, generate a key K_{ijs} and distribute this key to every member of the partition except s. Repeating this KPS for all partitions results in the subscribers of this BES receiving the keys according to Table 3.1.3.

1	2	3	4	5
K_{110}	K_{110}	K_{110}		
K_{112}	K_{111}	K_{111}		
K_{113}	K_{113}	K_{112}		
			K_{120}	K_{120}
			K_{125}	K_{124}
K_{210}	K_{210}		K_{210}	
K_{212}	K_{211}		K_{211}	
K_{214}	K_{214}		K_{212}	
		K_{220}		K_{220}
		K_{225}		K_{223}
K_{310}		K_{310}	K_{310}	
K_{313}		K_{311}	K_{311}	
K_{314}		K_{314}	K_{313}	
	K_{320}			K_{320}
	K_{325}			K_{322}

Table 3.1.3: Fiat-Naor Key Distribution Pattern for $PHF(3; 5, 2, 2)$

Once the pre-distribution phase has been completed, the broadcast phase is performed as follows. Let P be the privileged subset of subscribers and let $L_{ij} = \{s \mid s \in p_{ij},\ s \notin P\}$ be the set of columns in partition p_{ij} that are disjoint from

P. Now, let $K_{L_{ij}}$ represent the result of xor composition of all K_{ijl} where $l \in L_{ij}$. For broadcast message M, randomly generate message components $M_1, \ldots, M_N$ so that $\bigoplus_{i=1}^{N} M_i = M$. Then, for every partition p_{ij}, the final computed values will be the set of messages: $Y_{ij} = M_i \oplus K_{ij0} \oplus K_{L_{ij}}$

$$Y_{11} = M_1 \oplus K_{110} \oplus K_{112} \oplus K_{113}$$

$$Y_{12} = M_1 \oplus K_{120} \oplus K_{125}$$

$$Y_{21} = M_2 \oplus K_{210} \oplus K_{212}$$

$$Y_{22} = M_2 \oplus K_{220} \oplus K_{223} \oplus K_{225}$$

$$Y_{31} = M_3 \oplus K_{310} \oplus K_{313}$$

$$Y_{32} = M_3 \oplus K_{320} \oplus K_{322} \oplus K_{325}$$

Figure 3.1.4: Final Broadcast Content for BES on $PHF(3;\ 5, 2, 2)$

The concatenation of all Y_{ij} is the encrypted value for broadcast message M. In the $(5,2) - BES$ constructed above, broadcasting message M with a privileged subset $P = \{1, 4\}$, the broadcast would then be the concatenation of the encrypted components in Figure 3.1.4.

3.2 Introducing Scattering Hash Families

The core motivation behind this entire thesis is the following series of questions: If a $PHF(N; k, v, t)$ can be used to inflate a 1-resilient KPS into a t-resilient KPS that is efficient in both broadcast and key material overhead, what kind of construction can be used to inflate $KPSs$ of resilience $w > 1$? What type of combinatorial structure assures this property while still offering strong performance in both broadcast length and key material storage? And finally, what

are the advantages and disadvantages of such a scheme over existing BES techniques?

After analyzing the gamut of hash family variations, it was determined that no existing hash family met this criteria. Among the variations, $PHFs$ were the closest, providing the same security as with the Fiat-Naor BES but with significantly increased key material overhead. Since $PHFs$ offer tighter combinatorial restrictions on the t-subsets to inflate weaker $KPSs$, this result is as expected. Separating Hash Families ($SHFs$), Distributing Hash Families ($DHFs$), and their variants weaken the separation condition for $PHFs$ in such a way as to violate the unconditional security of the inflated BES. Specifically, this is due to both variants addressing partitions of t-subsets and only enforcing separation requirements between classes of partitions. Since no restriction is placed upon the relationship between the elements within a partition itself, the KPS construction loses its guarantee of separation.

It then remains to determine a generalization of $PHFs$ that takes advantage of the higher strength of ingredient schemes yet still offers the desired security under the provided construction model. Rather than partitioning t-subsets and redefining separation conditions based on these partitions, this generalization needs to enforce a variable multiplicity cap on the symbols in each subset. With these parameters in consideration, Scattering Hash Families ($ScHFs$) are defined.

Formally, a $(N; k, v, w, t) - Scattering\ Hash\ Family$ is a set of functions F such that $\forall f \in F$:

$$f: \{1, \dots, k\} \to \{1, \dots, v\}$$

and for any subset $C \subseteq \{1, \dots, k\}$ with $|C| = t$, $\exists f \in F$ such that for each symbol $s \in \{1, \dots, v\}$, the image $f(C)$ maps to the symbol s at most w times. Following the chosen notation for $PHFs$, Scattering Hash Families are represented as $ScHF(N, k, v, w, t)$. The term "Scattering" was chosen to convey the relaxation in separation requirements from a PHF. Within a t-subset, elements can clump together to a certain degree, but overall they need to be scattered fairly uniformly.

When viewed as an array, a $(N;\ k, v, w, t) - Scattering\ Hash\ Family$ is an $N \times k$ array populated with symbols from V where $|V| = v$ in such a way that for any selection of t columns of the matrix, there exists at least one row such that the symbols contained in the intersection of this row and the selected columns appear w or fewer times. The construction and combinatorial properties of $ScHFs$ are covered in detail throughout Sections 4.1 and 4.2 and the scalable, ingredient independent $ScHF\ BES$ is the primary focus of the following section.

Within a $ScHF(N; k, v, w, t)$, let $|s_i|$ represent the number of occurrences of symbol s in row i for every symbol $s \in \{1, \dots, v\}$. When, for all s, $1 \leq i \leq N$, $1 \leq j \leq N$, $|s_i| = |s_j|$, this $ScHF$ is called homogeneous. This is primarily an application-driven definition placed upon this hash family, which is further explored in Section 3.6 and utilized in the construction of a $ScHF\ BES$ in Section 3.4.

3.3 A Broadcast Encryption Scheme Based on $ScHFs$

As described in the previous section, $ScHFs$ were designed for the purpose of constructing $BESs$ from ingredient $KPSs$ of strength $w > 1$ while taking advantage of the increased ingredient strength in order to loosen the combinatorial

restriction on the hash family that determines the final key pre-distribution. $ScHFs$ achieve this by generalizing the t-subset separation condition from requiring complete element distinction to enforcing a multiplicity cap w per element. The construction of the $ScHF\ BES$ draws on the techniques used in the Fiat-Naor BES construction [1] and throughout this thesis, all $ScHF\ BESs$ constructed and analyzed utilize multiple instances of this construction for the ingredient $KPSs$. It is important to note that, by its definition, the scheme itself is agnostic to the ingredient $KPSs$ used in construction. Any w-resilient KPS can be used in place of the Fiat-Naor schemes used in this work. The choice to analyze only Fiat-Naor ingredient schemes was made to provide an initial scope for the analysis of the properties of this scheme.

Given a $ScHF(N;\ k, v, w, t)$, the central TA can construct a t-resilient BES with $k = |K|$ subscribers from w-resilient ingredient $KPSs$ as follows. For all rows of the $ScHF$, partition each row by the symbols present. Let p_{ij} be the label for the j^{th} partition in the i^{th} row of the $ScHF$. For each partition p_{ij}, construct a w-resilient KPS and deploy the symmetric keys accordingly. The $ScHF\ BES$ is generalized in such a way that no specific type of KPS is required for this stage and, moreover, the ingredient $KPSs$ do not need to be the of same type, so long as they are all w-resilient. For the sake of selection, however, construct a w-resilient KPS in the fashion described in Section 3.1. Now, for all p_{ij} construct a $PHF(N_{ij}; |p_{ij}|, v_{ij}, w)$ and from this, construct a Fiat-Naor KPS. Then, for all p_{ij} and for each subscriber $s \in p_{ij}$, distribute symmetric keys according to this KPS.

Once the key pre-distribution method has been deployed, the TA uses the following broadcast protocol. Let M be the message being broadcasted and let $P \in \mathcal{P}(K)$ be the privileged subset of subscribers for this broadcast. Beginning

with the same technique described in Section 3.1, generate $M_1, \dots, M_N$ random component messages of length $|M|$ such that

$$M = M_1 \oplus M_2 \oplus \dots \oplus M_N$$

where N is the number of rows of the $ScHF$. Instead of directly encrypting each of these message components, the $ScHF\ BES$ breaks down the message once more and encrypts each sub-component according to each partition's Fiat-Naor scheme. To do so, generate $M_{i1}, M_{i2}, \dots, M_{iN_{ij}}$ for all $M_i,\ 1 \le i \le N$, such that

$$M_i = M_{i1} \oplus M_{i2} \oplus \dots \oplus M_{iN_{ij}}.$$

For every partition p_{ij} of the $ScHF$, consider the $PHF(N_{ij}; |p_{ij}|, v_{ij}, w)$ that was constructed and deployed as a KPS on this partition. Within this PHF, retain the original $ScHF$ column indexing for continuity of representation. Let p_{ab} denote the b^{th} partition of the a^{th} row of the PHF and let $L_{ab} = \{s \mid s \in p_{ab},\ s \notin P\}$ be the set of columns in partition p_{ab} that are disjoint from P. Now, let $K_{ijL_{ab}}$ represent the xor composition of all K_{ijabl} where $l \in L_{ab}$. For each partition p_{ab}, construct the set of encrypted message components: $Y_{ijab} = M_{ia} \oplus K_{ijab0} \oplus K_{ijL_{ijab}}$. For all $ScHF$ partitions p_{ij} and for all PHF partitions p_{ab} corresponding to each p_{ij}, the encrypted broadcast is the concatenation of all Y_{ijab}:

$$p_{11}\colon Y_{11ab} = M_{1a} \oplus K_{ab0} \oplus K_{11L_{11ab}}.$$

$$\vdots$$

$$p_{12}\colon Y_{12ab} = M_{2a} \oplus K_{ab0} \oplus K_{12L_{12ab}}$$

$$\vdots$$

$$p_{Nv}\colon Y_{Nvab} = M_{Na} \oplus K_{ab0} \oplus K_{NvL_{NN_{ij}ab}}$$

Figure 3.3.1: The Final Broadcast Content for a $ScHF\ BES$

The formal definition of the $ScHF\ BES$ is given in Figure 3.3.2 and Figure 3.3.3. This is the scheme that is used for the direct comparison with the Fiat-Naor $PHF\ BES$ in Section 5.3. As a step towards producing that comparison, a means of computing the efficiency of a BES by measuring its information rate is introduced.

Pre-distribution Phase:

- Let K be the set of all subscribers of the BES with $|K| = k$ and let $S = ScHF(N; k, v, w, t)$.
- For every row R_i, $1 \leq i \leq N$, partition the set of columns by symbol into partitions p_{ij}, which represents the j^{th} partition of row R_i.
- For each partition p_{ij}, construct a $PHF(N_{ij}; |p_{ij}|, v_{ij}, w)$ and from this, construct a Fiat-Naor KPS. For all p_{ij} and for each subscriber $s \in p_{ij}$, distribute keys according to this KPS, preserving the original column indexing.

Figure 3.3.2: Pre-distribution Phase of a $ScHF$-Based BES

Broadcast Phase:

- For broadcast message M, let $P \subseteq K$ be the privileged subset.
- Construct component messages $M_1, M_2, \dots, M_N$ of size $|M|$ uniformly at random such that $M = M_1 \oplus M_2 \oplus \dots \oplus M_N$.
- For every $ScHF$ partition p_{ij}, consider the $PHF(N_{ij}; |p_{ij}|, v_{ij}, w)$ deployed on this partition, but retain the original $ScHF$ column indexing:
    - The broadcast message for this partition is M_i, construct components $M_{11}, \dots, M_{1N_{ij}}$ such that $M_i = M_{11} \oplus \dots \oplus M_{1N_{ij}}$.
    - Let p_{ab} denote the b^{th} partition of the a^{th} row of the PHF and let $L_{ab} = \{s \mid s \in p_{ab},\ s \notin P\}$ Now, let $K_{L_{ab}}$ represent $\oplus K_{abl}$ $\forall l \in L_{ab}$.
    - For each partition p_{ab}, construct the set of encrypted message components: $Y_{ab} = M_{ia} \oplus K_{ab0} \oplus K_{L_{ab}}$.
- For all $ScHF$ partitions p_{ij} and for all PHF partitions p_{ab} corresponding to each p_{ij}, broadcast $Y_{ijab} = M_{ia} \oplus K_{ijab0} \oplus K_{ijL_{ab}}$.

Figure 3.3.3: Broadcast Phase of a $ScHF$-Based BES

When designing Broadcast Encryption Schemes, it is useful to be able to meaningfully compare them to one another in terms of their performance. The information rate of a BES is one such metric [16]. The information rate can be framed as the efficiency with which a BES distributes secret information to a user. Formally, the information rate of a BES is defined as

$$\rho = \min\left\{\frac{\log q}{H(U_i)} : 1 \le i \le k\right\}$$

where U_i is the set of all possible secret values that can be distributed to subscriber i, and $H(U_i)$ is the familiar entropy function, which measures the uncertainty associated with the random variable U_i. Alternatively, as formalized by Shannon [24], this value is measuring the information content missing by not knowing value of the random variable. In this definition, it is assumed that all keys $K_i \in GF(q)$, the Galois Field of prime or prime power order q. For this reason, $H(K_P)$ has already been reduced to $\log q$ in the equation. In this situation, the actual key selection method is irrelevant because the information rate being computed is the ratio between two key entropies and therefore, the entropic value specific to the key selection cancels out.

When constructed upon a $PHF(N; k, v, t)$, the information rate of a Fiat-Naor BES has been shown to be:

$$\rho = \frac{1}{Nk}$$

which is directly calculable from the parameters of the PHF deployed. In this scheme, the amount of secret information being delivered is $H(K_P)$, where K_P is the key associated with the privileged subset of a given broadcast. In order to obtain this, the scheme distributes $\frac{k}{v}$ keys to every user per partition of the PHF. Since there are $N * v$ partitions, the information content of the distributed key material is:

$$N * v * \frac{k}{v} * H(K_i) = Nk * H(K_i)$$

And thus, the ratio of these information contents results in the rate:

$$\rho = \frac{H(K_P)}{NkH(K_i)} = \frac{1}{Nk}$$

Now, to determine the information rate of the $ScHF\ BES$, it remains to formulate the amount of distributed key material is required by this scheme. On a $ScHF(N_1; k_1, v_1, t_1)$ with an ingredient KPS built on $PHF(N_2; k_2, v_2, t_2)$, there will be $N_1 v_1 (N_2 v_2 k_2 / v_2)$ keys distributed to each subscriber. That is, for all of the $N_1 v_1$ $ScHF$ partitions, each user receives the amount of key material defined in the Fiat-Naor Scheme above, specifically $N_2 v_2 k_2 / v_2$ keys per KPS. Due to the relationships between the parameters of this $ScHF$ and PHF as enforced by this scheme, the size of these families are related by $k_2 = k_1 / v_1$ when v_1 is chosen minimally. The information rate for the $ScHF\ BES$ is then:

$$\rho = \frac{H(K_P)}{N_1 N_2 k_1 H(K_i)} = \frac{1}{N_1 N_2 k_1}$$

This value will be used as one means of measuring the storage efficiency between the PHF-based $BESs$ and the $ScHF$-based $BESs$. The inverse value of this ratio is defined as the Key Material Overhead (KMO), which gives an indication as to the number of keys distributed to each user in order to determine the feasibility of deploying such a scheme. As demonstrated by the two naïve $BESs$ in Section 2.2, this value alone is not enough to determine the quality of a BES. Complementary to the information rate is the Broadcast Encryption Overhead (BEO) and both of these metrics are utilized extensively in Chapter 5 to form a meaningful comparison between the two BES designs.

3.4 The Construction of a $ScHF\ BES$

Consider the construction of a Broadcast Encryption Scheme on $k = 10$ subscribers with resilience against colluding parties of size at most $t = 3$. For this scheme, any ingredient KPS of resilience $1 \leq w \leq t$ may be chosen,

however, $w = 1$ restricts the $ScHF$ into a PHF and $w = t$ is a trivial $ScHF$ of one row, as demonstrated in Section 3.6. For this reason, a $ScHF(3; 10, 2, 2, 3)$ is a reasonable choice in this scheme. This family is a minimal $ScHF$ matching the dimensions and strength requirements for the BES desired. The following is a construction of a $(10,3) - BES$ utilizing the $ScHF\ BES$ as defined in Figure 3.3.2.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	2	2	1	1	2	2	1	1	2
1	1	1	2	2	1	1	2	2	2
1	1	1	1	1	2	2	2	2	2

Table 3.4.1: A $ScHF(3; 10, 2, 2, 3)$

After generating the $ScHF(3; 10, 2, 2, 3)$, partition the column indices of every row $R_i,\ 1 \leq i \leq N$, based on the symbol appearing at that location. Let p_{ij} be the label for the j^{th} partition in row R_i. Note that in this $ScHF$, for every p_{ij}, $|p_{ij}| = 5$. In this instance, this phenomenon is due to the fact that the selected $ScHF$ is homogeneous, allowing the construction of a $ScHF\ BES$ from a single ingredient KPS. In general, all partitions are not required to be of the same size; however, $ScHF\ BESs$ lacking this property require the construction of multiple ingredient $KPSs$ to apply to the partitions of each dimension present.

$p_{11} = \{1, 4, 5, 8, 9\}$	$\{2, 3, 6, 7, 10\} = p_{12}$
$p_{21} = \{1, 2, 3, 6, 7\}$	$\{4, 5, 8, 9, 10\} = p_{22}$
$p_{31} = \{1, 2, 3, 4, 5\}$	$\{6, 7, 8, 9, 10\} = p_{32}$

Table 3.4.2: Symbolic Partitions for $ScHF(3; 10, 2, 2, 3)$

On every partition p_{ij}, create a w-resilient KPS on k_{ij}. Let each of these be a $(|p_{ij}|, w)$-Fiat-Naor KPS. Each instance of these $KPSs$ needs to be constructed from a $PHF(N_{ij}, |p_{ij}|, w, w)$, which has precisely the parameters of the PHF and KPS constructed in Section 3.1. For each $ScHF$ partition p_{ij}, and for each partition p_{ab} of the PHF applied to p_{ij}, distribute the set of keys K_{ijabs} where $s \in \{1, \dots, v\} \cup \{0\}$ to the subscribers in accordance with each KPS. The key distribution pattern is given in Figures 3.4.3, 3.4.4, and 3.4.5. For broadcast message M and a privileged subset of subscribers P, the explicit final broadcast is given in Figure 3.4.2.

$$M \begin{cases} M_1 \begin{cases} Y_{1111} = M_{11} \oplus K_{11110} \oplus K_{11L_{1111}} \\ Y_{1112} = M_{11} \oplus K_{11120} \oplus K_{11L_{1112}} \\ Y_{1121} = M_{12} \oplus K_{11210} \oplus K_{11L_{1121}} \\ Y_{1122} = M_{12} \oplus K_{11220} \oplus K_{11L_{1122}} \\ Y_{1131} = M_{13} \oplus K_{11310} \oplus K_{11L_{1131}} \\ Y_{1132} = M_{13} \oplus K_{11320} \oplus K_{11L_{1132}} \end{cases}, \begin{cases} Y_{1211} = M_{11} \oplus K_{12110} \oplus K_{12L_{1211}} \\ Y_{1212} = M_{11} \oplus K_{12120} \oplus K_{12L_{1212}} \\ Y_{1221} = M_{12} \oplus K_{12210} \oplus K_{12L_{1221}} \\ Y_{1222} = M_{12} \oplus K_{12220} \oplus K_{12L_{1222}} \\ Y_{1231} = M_{13} \oplus K_{12310} \oplus K_{12L_{1231}} \\ Y_{1232} = M_{13} \oplus K_{12320} \oplus K_{12L_{1232}} \end{cases} \\ M_2 \begin{cases} Y_{2111} = M_{21} \oplus K_{21110} \oplus K_{21L_{2111}} \\ Y_{2112} = M_{21} \oplus K_{21120} \oplus K_{21L_{2112}} \\ Y_{2121} = M_{22} \oplus K_{21210} \oplus K_{21L_{2121}} \\ Y_{2122} = M_{22} \oplus K_{21220} \oplus K_{21L_{2122}} \\ Y_{2131} = M_{23} \oplus K_{21310} \oplus K_{21L_{2131}} \\ Y_{2132} = M_{23} \oplus K_{21320} \oplus K_{21L_{2132}} \end{cases}, \begin{cases} Y_{2211} = M_{21} \oplus K_{22110} \oplus K_{22L_{2211}} \\ Y_{2212} = M_{21} \oplus K_{22120} \oplus K_{22L_{2212}} \\ Y_{2221} = M_{22} \oplus K_{22210} \oplus K_{22L_{2221}} \\ Y_{2222} = M_{22} \oplus K_{22220} \oplus K_{22L_{2222}} \\ Y_{2231} = M_{23} \oplus K_{22310} \oplus K_{22L_{2231}} \\ Y_{2232} = M_{23} \oplus K_{22320} \oplus K_{22L_{2232}} \end{cases} \\ M_3 \begin{cases} Y_{3111} = M_{31} \oplus K_{31110} \oplus K_{31L_{3111}} \\ Y_{3112} = M_{31} \oplus K_{31120} \oplus K_{31L_{3112}} \\ Y_{3121} = M_{32} \oplus K_{31210} \oplus K_{31L_{3121}} \\ Y_{3122} = M_{32} \oplus K_{31220} \oplus K_{31L_{3122}} \\ Y_{3131} = M_{33} \oplus K_{31310} \oplus K_{31L_{3131}} \\ Y_{3132} = M_{33} \oplus K_{31320} \oplus K_{31L_{3132}} \end{cases}, \begin{cases} Y_{3211} = M_{31} \oplus K_{32110} \oplus K_{32L_{3211}} \\ Y_{3212} = M_{31} \oplus K_{32120} \oplus K_{32L_{3212}} \\ Y_{3221} = M_{32} \oplus K_{32210} \oplus K_{32L_{3221}} \\ Y_{3222} = M_{32} \oplus K_{32220} \oplus K_{32L_{3222}} \\ Y_{3231} = M_{33} \oplus K_{32310} \oplus K_{32L_{3231}} \\ Y_{3232} = M_{33} \oplus K_{32320} \oplus K_{32L_{3232}} \end{cases} \end{cases}$$

Figure 3.4.3: Final Broadcast Content for $(10,3)$-$ScHF\ BES$

1	2	3	4	5	6	7	8	9	10
K_{11110} K_{11114} K_{11115}			K_{11110} K_{11111} K_{11115}	K_{11110} K_{11111} K_{11114}					
`							K_{11120} K_{11129}	K_{11120} K_{11128}	
K_{11210} K_{11214} K_{11218}			K_{11210} K_{11211} K_{11218}				K_{11210} K_{11211} K_{11214}		
				K_{11220} K_{11229}				K_{11220} K_{11225}	
K_{11310} K_{11315} K_{11318}				K_{11310} K_{11311} K_{11318}			K_{11310} K_{11311} K_{11315}		
			K_{11320} K_{11329}					K_{11320} K_{11324}	
	K_{12110} K_{12113} K_{12116}	K_{12110} K_{12112} K_{12116}			K_{12110} K_{12112} K_{12113}				
						K_{12120} K_{121210}			K_{12120} K_{12127}
	K_{12210} K_{12213} K_{12217}	K_{12210} K_{12212} K_{12217}				K_{12210} K_{12212} K_{12213}			
					K_{12220} K_{122210}				K_{12220} K_{12226}
	K_{12310} K_{12316} K_{12317}				K_{12310} K_{12312} K_{12317}	K_{12310} K_{12312} K_{12316}			
		K_{12320} K_{123210}							K_{12320} K_{12323}

Table 3.4.4: $ScHF\ BES$ Key Pre-distribution for p_{11} and p_{12}

1	2	3	4	5	6	7	8	9	10
K_{21110}	K_{21110}	K_{21110}							
K_{21112}	K_{21111}	K_{21111}							
K_{21113}	K_{21113}	K_{21112}							
					K_{21120}	K_{21120}			
					K_{21127}	K_{21126}			
K_{21210}	K_{21210}				K_{21210}				
K_{21212}	K_{21211}				K_{21211}				
K_{21216}	K_{21216}				K_{21212}				
		K_{21220}				K_{21220}			
		K_{21227}				K_{21223}			
K_{21310}		K_{21310}			K_{21310}				
K_{21313}		K_{21311}			K_{21311}				
K_{21316}		K_{21316}			K_{21313}				
	K_{21320}					K_{21320}			
	K_{21327}					K_{21322}			
			K_{22110}	K_{22110}			K_{22110}		
			K_{22115}	K_{22114}			K_{22114}		
			K_{22118}	K_{22118}			K_{22115}		
								K_{22120}	K_{22120}
								K_{221210}	K_{22129}
			K_{22210}	K_{22210}				K_{22210}	
			K_{22215}	K_{22214}				K_{22214}	
			K_{22219}	K_{22219}				K_{22215}	
							K_{22220}		K_{22220}
							K_{222210}		K_{22228}
			K_{22310}				K_{22310}	K_{22310}	
			K_{22318}				K_{22314}	K_{22314}	
			K_{22319}				K_{22319}	K_{22318}	
				K_{22320}					K_{22320}
				K_{223210}					K_{22325}

Table 3.4.5: $ScHF\ BES$ Key Pre-distribution for p_{21} and p_{22}

1	2	3	4	5	6	7	8	9	10
K_{31110}	K_{31110}	K_{31110}							
K_{31112}	K_{31111}	K_{31111}							
K_{31113}	K_{31113}	K_{31112}							
			K_{31120}	K_{31120}					
			K_{31125}	K_{31124}					
K_{31210}	K_{31210}		K_{31210}						
K_{31212}	K_{31211}		K_{31211}						
K_{31214}	K_{31214}		K_{31212}						
		K_{31220}		K_{31220}					
		K_{31225}		K_{31223}					
K_{31310}		K_{31310}	K_{31310}						
K_{31313}		K_{31311}	K_{31311}						
K_{31314}		K_{31314}	K_{31313}						
	K_{31320}			K_{31320}					
	K_{31325}			K_{31322}					
					K_{32110}	K_{32110}	K_{32110}		
					K_{32117}	K_{32116}	K_{32116}		
					K_{32118}	K_{32118}	K_{32117}		
								K_{32120}	K_{32120}
								K_{321210}	K_{32129}
					K_{32210}	K_{32210}		K_{32210}	
					K_{32217}	K_{32216}		K_{32216}	
					K_{32219}	K_{32219}		K_{32217}	
							K_{32220}		K_{32220}
							K_{322210}		K_{32228}
					K_{32310}		K_{32310}	K_{32310}	
					K_{32318}		K_{32316}	K_{32316}	
					K_{32319}		K_{32319}	K_{32318}	
						K_{32320}			K_{32320}
						K_{323210}			K_{32327}

Table 3.4.6: $ScHF\ BES$ Key Pre-distribution for p_{31} and p_{32}

3.5 Unconditional Security of the $ScHF$-Based BES

As demonstrated in Section 3.3, the broadcast message M is broken up into component messages $M_1, \ldots, M_N$ and each component is encrypted and broadcast in such a way as to deter an adversary or colluding party of adversaries from collecting enough pieces to reassemble M. In this section, the proof of security for $ScHF$-based $BESs$ is given in detail.

To begin, assume an adversary is able to recover all M_i for $1 \leq i \leq N$. This adversary will be able to trivially recover M by computing the xor of all these values. Now, assume an adversary is only able to recover $N-1$ component messages from $M_1, \ldots, M_N$. Without loss of generality, let the missing component be M_N. Let

$$M_R = M_1 \oplus M_2 \oplus \ldots \oplus M_{N-1}$$

denote the exclusive-or of the recoverable components of M. The equation for M can then be simplified to $M = M_R \oplus M_N$. Exclusive-oring both sides by M_N results in $M \oplus M_N = M_R$, the familiar form of the One-Time Pad encryption scheme in which M is the plaintext, M_N is the secret random key of bit length $|M|$, and M_R is the exposed ciphertext. Following the famous information-theoretic results of Shannon [24], this scheme has the property that the entropy of message M given the possession of ciphertext M_R is exactly equal to the entropy of message M itself, represented traditionally as $H(M|C) = H(M)$. The implications of this property in this BES construction are that not only can an adversary not recover M in its entirety when recovering at most $N-1$ component messages, but also having done so reveals not even one bit of information about M to the adversary. In order to demonstrate the security of this $ScHF\ BES$ construction, it now remains to show that the scheme ensures that any adversary or colluding party

of adversaries of at most t members can recover no more than $N-1$ component messages.

For every partition p_{ij} of the $ScHF$, message component M_i is encrypted in the following fashion: p_{ij}: $Y_{ab} = M_{ia} \oplus K_{ab0} \oplus K_{L_{ab}}$ where $K_{L_{ab}}$ represents the xor composition of all K_{abl} where $l \in L_{ab}$ and the tuple (a, b) iterates over all partitions of the PHF applied to the $(i,j)^{th}$ $ScHF$ partition. By encrypting each message sub-component with the keys that the disjoint set L_{ab} are missing, each KPS protects against colluding parties of size at most w. Now, since the $ScHF$ splits each message component in such a way that some row ensures that no more than w members of a colluding party are in the same partition for any choice of t colluding partners, this row prevents the recovery of one M_{ia}. As demonstrated in Section 3.1, in order for an adversary to obtain M, it is necessary to obtain all $M_1, \dots, M_N$. In turn, since these values are merely the xor composition of all $M_{i1}, \dots, M_{iN_{ij}}$, the following is the computation for M:

$$M = M_{11} \oplus \dots \oplus M_{1N_{ij}} \oplus M_{21} \oplus \dots \oplus M_{2N_{ij}} \oplus \dots \oplus M_{N1} \oplus \dots \oplus M_{NN_{ij}}$$

Thus, in order to obtain M, an adversary must recover all M_{ia}, $1 \leq i \leq N$, $1 \leq a \leq N_{ij}$ [24]. It has been shown that there exists at least one M_{ia} that a colluding party of size t or smaller cannot obtain and as such, this scheme is secure. While some broadcast encryption methods offer some probabilistic security against colluding parties greater than the defined resilience [7], this scheme is immediately broken upon the $(t+1)^{th}$ member entering the colluding party. It is a necessary restriction that no more than t columns be chosen since the $ScHF$ does not guarantee separation for subsets of any larger size, even if many such subsets are indeed separated.

3.6 Properties of $ScHF$-Based $BESs$

When starting with a fixed $(k_1, t_1) - KPS$, constructing a $ScHF\ BES$ from this KPS will obey the properties listed below. As introduced in Section 3.2, the $ScHF$ utilized in this construction will necessarily be homogeneous. Let the final $ScHF$ be represented by S: $ScHF(N; k, v, w, t)$.

Lemma 3.6.1: When a $(k, t) - ScHF\ BES$ is constructed from a single fixed ingredient $(k_1, t_1) - KPS$ and utilizing a homogeneous $ScHF(N; k, v, w, t)$, then $k_1 | k$.

Lemma 3.6.2: When a $(k, t) - ScHF\ BES$ is constructed from a single fixed ingredient $(k_1, t_1) - KPS$ and utilizing a homogeneous $ScHF(N; k, v, w, t)$, then$t_1 \geq w$

Since the ingredient KPS is fixed by choice, Lemma 3.6.1 is a direct result of the homogenous ingredient restriction. Each row contains k columns which must be able to be partitioned into an integral number of partitions of size k_1, thus $k_1 | k$.

Lemma 3.6.2 follows from the security property of the $ScHF$ scheme. By $ScHF$ definition, for each t-subset T of S, there must exist at least one row in which no symbol appears more than w times. Accordingly, the $KPSs$ applied to each partition must be resilient against colluding parties of size at least w.

Assume to the contrary that $t_1 < w$. Let R_i represent the row of S that guarantees separation for the t-subset T_i. If any of the $ScHF$ elements located at $R_i \cap T_i$ are repeated w times, as allowed by the parameters of S, the KPS applied

to the partition containing these elements does not secure against colluding parties of size w and as such, the message component encrypted by this partition is no longer provably unrecoverable. Using the notation presented in Section 3.5, this message component corresponds to M_N, the singular message component that is required to be unrecoverable to prevent recovery of M. The assumption $t_1 < w$ thus violates the security property of the $ScHF\ BES$ proven Section 3.5 and so it must be that $t_1 \geq w$.

Chapter 4

CONSTRUCTING SCATTERING HASH FAMILIES

4.1 Theoretic Results on the Existence of $ScHFs$

Following are theorems pertaining to the existence of Scattering Hash Families for various restrictions of parameters. A formalization of the PHF generalization claims for $ScHFs$ is given in Theorem 4.1.1. As was proven for $PHFs$ [3], there exist certain conditions under which a $ScHF$ can be trivially constructed in one row. Theorem 4.1.2 provides bounds for the existence of these trivial $ScHFs$ which are demonstrated via constructive proof. Theorems 4.1.3 and 4.1.4 provide some introductory insight into the existence of minimal $ScHFs$ under specific sets of parameters. Finally, Theorem 4.1.5 is a $ScHF$-specific result that follows from the work of Stein [25], Lovász [26], and Johnson [27] that lay the framework for generalized hash family bounds. This theorem relies on properties of randomly generated arrays and is the primary focus of the derandomization in Section 4.2.

Theorem 4.1.1: $PHFs$ are a subclass of $ScHFs$. By enforcing a multiplicity cap of $w = 1$, the separation condition for a $ScHF$ enforces every t-subset to be mapped to distinct symbols in some row of the structure. As such, any construction of a PHF is a construction of a $ScHF$ with $w = 1$.

Proof: Let $S = ScHF(N; k, v, 1, t)$ be a Scattering Hash Family. By definition, S is a set of functions F such that $\forall f \in F$:

$$f: \{1, \dots, k\} \to \{1, \dots, v\}$$

and for any subset $C \subseteq \{1, \dots, k\}$ with $|C| = t$, $\exists f \in F$ such that for each symbol $s \in \{1, \dots, v\}$, the image $f(C)$ maps to the symbol s at most w times. In this

instance, $w = 1$ and accordingly, the image of any t-subset C is a set of elements whose symbols are repeated at most once. Therefore, all elements of this image are distinct, providing the necessary separation condition for S to be a $PHF(N; k, v, t)$. □

Theorem 4.1.2: When $w * v \geq k$ or $w \geq t$, a $ScHF$ exists on 1 row.

Proof: Let A be a sequence of elements a_i from the set $a_i \in \{1, \dots, v\}$. For some $w \geq 1$, populate A in the following fashion:

$$A = \{1, \dots, 1_w, 2, \dots, 2_w, \dots, v, \dots, v_w\}$$

This array contains $v * w$ elements, in which no element is repeated more than w times. Trivially, no subset $A_k \subseteq A$ of size $k \leq v * w$ has any symbol repeated more than w times. Moreover, for any subset of size t of A_k, no symbol can be repeated more than w times as well. This construction is therefore a $ScHF(1; k, v, w, t)$. Additionally, for any subset of size t, it is impossible for any symbol to repeated more than w times, so for all $w \geq t$, the same construction applies. □

Theorem 4.1.3: If a $ScHF(N; k, v, w, t)$ exists where $t > 1$, then there is some $N_2 \leq N$ for which a $ScHF(N_2; k, v, w, t - 1)$ exists.

Proof: Let $S_1 = ScHF(N; k, v, w, t)$ be a Scattering Hash Family. By the array definition of $ScHFs$, this family is a $N \times k$ array and has the property that for any t-subset, there exists a row such that the symbols appearing in this row and the column indices appear no more than w times. Let $T = \{a_1, \dots, a_t\}$ be the set of

elements appearing in one such t-subset. Let s be the symbol with the highest occurrence n_s among elements a_i. Now, for any subset T' of these elements of size $t' = t - 1$, where $t' \geq 1$, either $n'_s = n_s - 1$ in the case that s is the unique symbol of highest frequency and a copy of s was removed, or $n'_s = n_s$ in the case that a different symbol was removed and s is therefore still the most frequent. Since, by definition of S_1, $n_s \leq w$, it is the case that $n'_s \leq n_s \leq w$ and as such, there exists some $S_2 = ScHF(N_2; k, v, w, t-1)$. In this instance, S_1 provides existence for $N_2 = N$, however, since the size of subsets being considered has decreased, it may be possible to express S_2 in fewer rows, thus $N_2 \leq N$. □

Theorem 4.1.4: If a $ScHF(N; k, v, w, t)$ exists, then there is some $N_2 \leq N$ for which a $ScHF(N_2; k, v, w+1, t)$ exists.

Proof: The proof of this follows that of Theorem 4.1.3. Let $S_1 = ScHF(N; k, v, w, t)$ be a Scattering Hash Family. S_1 is an $N \times k$ array of symbols from the set $\{1, \dots, v\}$. Let $T = \{a_1, \dots, a_t\}$ be the set of elements appearing in any t-subset of columns. Let s be the symbol with the highest occurrence n_s among elements a_i. The inequality $n_s \leq w$ is necessary for S_1 to be a $ScHF$ and so it is trivially the case that $n_s \leq w + 1$. Since this holds for any t-subset of elements in the array, S_1 is therefore a $ScHF(N_2;\ k, v, w+1, t)$. As with Theorem 4.1.3, this existence is proven for $N_2 = N$, but a smaller instance may exist due to the relaxed separation requirements and so $N_2 \leq N$. □

In Section 3.3, the method was shown for constructing $BESs$ from $ScHFs$, but it remains to demonstrate the existence of these hash families for reasonable

parameters. As an initial step in this effort, a non-constructive proof of existence for $ScHFs$ is provided in Theorem 4.1.5. The core property of this proof is that it harnesses the probabilistic separation of a randomly generated t-subset to determine an upper bound on the number of rows required to ensure that for some array of this size, all t-subsets are $ScHF$-separated and thus a $ScHF$ of these parameters necessarily exists. In order to determine the expected separation, the procedure $sep_{ScHF}(v, w, t)$ is defined as follows.

input: v: Size of the set of symbols
t: The number of positions being filled with symbols
w: Maximum number of times a symbol can appear

output: The number of ways that a set of t positions can be populated by symbols from $\{1, \dots, v\}$ in such a way that no symbol is repeated more than w times. Specifically, this is the number of ways a t-subset will meet the $ScHF$ separation condition.

procedure $sep_{ScHF}(v, w, t)$:
begin
    if $v = 0$ and $t = 0$ **then**
        return 0
    else if $v = 0$ **then**
        return 1
    else
        return $\sum_{i=0}^{\min(w,t)} \binom{t}{i} * sep_{ScHF}(v-1, w, t-i)$
end

Figure 4.1.1: Definition of $sep_{ScHF}(v, w, t)$ function

Theorem 4.1.5: For any nonnegative integers N, k, w, t, if:

$$1 > \binom{k}{t}\left(1 - \frac{sep_{ScHF}(v, w, t)}{v^t}\right)^N$$

then there exists a $ScHF(N;\ k, v, w, t)$. The function $sep_{ScHF}(v, w, t)$ is defined in Figure 4.1.1.

Proof: Let A be a $N \times k$ array consisting of symbols chosen uniformly at random from $V = \{1, \dots, v\}$. Consider any t-subset of columns $T \subseteq \{1, \dots, k\}$. Since the elements are chosen randomly, the expectation that this t-subset is separated in a given row is simply the number of ways to separate this t-subset divided by all possible row values. The procedure sep_{ScHF} is a recursive routine that computes the number of ways to arrange at most w copies of a symbol into t distinct positions, known commonly as the multiset coefficient. This is precisely the number of possible ways to $ScHF$-separate a subset of size t. There are v^t ways to populate this row, so the quantity $1 - \frac{sep_{ScHF}(v,w,t)}{v^t}$ is then the complement of the expectation of separation for a t-subset for a given row. More specifically, this is the expectation that this t-subset is not separated in one row. By raising this value to the N^{th} power, this value represents the expectation that a t-subset is not separated in N independent trials, which correspond the rows in this scenario since each element is generated at random. Multiplying this total expectation for an arbitrary t-subset by the total number of t-subsets, $\binom{k}{t}$, yields a bound on the expectation that no t-subset is separated. Since the number of separations is an integral value, if this expectation is less than 1, then it means that there exists some $N \times k$ array with v symbols that is a $ScHF(N; k, v, w, t)$. □

4.2 A Derandomized Construction of $ScHFs$

In order to derandomize the non-constructive $ScHF$ existence proof provided in Section 4.1, this technique is first formalized for the construction of $PHFs$. This result is used to generalize to an algorithm that, when incorporated with the proper separation condition, will deterministically construct a wide variety of hash

families. The separation conditions of $ScHFs$ are then applied to this general algorithm, resulting in a deterministic construction algorithm for $ScHFs$. The technique used to derandomize the probabilistic construction in this work is the Method of Conditional Expectation [20]. This method removes the random choice in a probabilistic proof or algorithm by computing the conditional expectation of success for each possible value of this choice and deterministically selecting a value among these that meets or exceeds the expectation of success. This approach for efficiently and deterministically constructing combinatorial structures using a density-based algorithm was first put forth for pairwise testing [28] and Covering Arrays [29] and subsequently demonstrated for $PHFs$ [30]. Since $ScHFs$ generalize $PHFs$, this technique is a natural choice for constructing initial bounds on this type of family.

In order to utilize the density-based algorithmic approach, it is first necessary to define the conditional expectation in terms of the choice being derandomized. For a PHF, the conditional expectation is the expected number of t-subsets newly separated by fixing an as-yet undetermined entry to a symbol $s \in \{1, \dots, v\}$. The formalization of this expected separation and subsequently, the conditional expected separation is as follows.

Let T be a t-subset of elements of a particular row R of a PHF consisting of fixed symbols and/or undetermined entries denoted by "$*$". Without loss of generality, the t-subset can be arranged as follows:

$$T = S_1, S_2, \dots, S_{t-d}, *_1, *_2, \dots, *_d$$

where $|T| = t$ and consists of d undetermined entries and $t - d$ fixed entries. In order to compute the expected separation of this t-subset, first check for separation condition violations. In the case of a PHF, a separation condition

violation is simply a collision: $\exists S_i, S_j \in T,\ i \neq j$ such that $S_i = S_j$. If a violation exists, the expected separation of this t-subset is necessarily 0 since no completion of this t-subset will prevent this violation. If there does not exist a collision in the populated elements, however, the computation of the expectation is as follows.

For the d remaining entries to be filled, there are v^d ways to fully populate the undetermined entries. Since it is known that all S_i are unique, there are $v-(t-d)$ ways to choose a valid symbol for $*_1$, $v-(t-d)-1$ ways to choose a valid symbol for $*_2$, and so on. Thus, the expectation that this subset be separated from all possible remaining completions of this subset is:

$$ES_{PHF}(T) = \frac{\prod_{i=1}^{d}(v+1-(t-d)-i)}{v^d}$$

In fact, this is shown to be a generalization of the formula for the probability that a t-subset T_* consisting entirely of undetermined entries is separated. In this situation, $d = t$ and the formula simplifies as follows:

$$ES_{PHF}(T_*) = \frac{\prod_{i=1}^{d}(v+1-(t-d)-i)}{v^d} = \frac{\prod_{i=1}^{t}(v+1-i)}{v^t}$$

Moreover, once this formula is obtained, the PHF analogue to the non-constructive $ScHF$ existence proof can be expressed. By the same argument as the proof of Theorem 4.1.5, if, for any nonnegative integers N, k, t, the following inequality holds:

$$1 > \binom{k}{t}(1-ES_{PHF}(T_*))^N$$

then there necessarily exists a $PHF(N;k,v,t)$. A naïve attempt at forming a construction from this proof of existence is to create an array matching a set of parameters satisfying the above inequality and populate the entire array uniformly at random. At each stage, check to see if the structure produced is a PHF and if not, generate it again until the separation conditions are satisfied. This approach is a randomized algorithm in the Las Vegas style since any output necessarily satisfies all separation constraints, however, the algorithm is not guaranteed to terminate.

Consider, for the purposes of derandomization, the Las Vegas style randomized algorithm used to construct instances of $ScHFs$ whose existence are proven by the inequality in Theorem 4.1.5. Instead of generating an entire row or even an entire array at random, consider the selection of a single element. If, for the selection of every element in the array, the expected number of separations is at least as high as before fixing this symbol, then as the row is filled, this expectation becomes the actual number of separations. As mentioned previously, this technique was formalized for other combinatorial structures by Colbourn and Bryce [28] [30] and is now modified to construct $ScHFs$.

Let A be an $N \times k$ array initially populated entirely with $*$, denoting undetermined entries. Select the leftmost entry of the topmost row that is still a $*$. For all ways of fix this to a symbol $s \in \{1, \dots, v\}$, calculate the conditional expected separations of fixing this entry to s. Rather than selecting the optimal choice at every point, selecting one that is as good as average will suffice. Thus, the expected separation of the row after each selection is at least as high as it was previously and as each row is completed, the actual number of t-subset separations increases with the same rate as theorized by the inequality from

Theorem 4.1.5 upon iterating over $1 \leq i \leq N$. When all N rows have been completed, the $ScHF$ has been constructed. The technique of reaching down and computing the expected separation for all possible choices for a given decision is precisely the desired approach for the Method of Conditional Expectation. Moreover, in this instance, it has been demonstrated to produce an efficient deterministic algorithm for actually constructing $ScHFs$ from what started as a purely non-constructive existential proof. This algorithm is presented in full detail in Figure 4.2.1.

Input: k: The number of columns for the $ScHF$
v: The size of the set of symbols
t: The size of subsets needing to be separated
w: The number of times a symbol can appear in a t-subset
Output: $ScHF(N; k, v, w, t)$

procedure $constructScHF(k, v, w, t)$:
begin
Initialize A to empty array
Initialize $unseparatedSubsets$ to $\binom{k}{t}$
while $unseparatedSubsets > 0$
Initialize $density$ to expected separation of current row state
Add row $\{*_1, \ldots, *_k\}$ to bottom of A
Initialize d to t
while there remain any $*$ in new row
$density :=$ expected number of t-subsets separated by randomly completing the remainder of this row
Select index of leftmost $*$ in row, call this $*_i$
for all v ways to fix this symbol to $s \in \{1, \ldots, v\}$
for all $\binom{k-1}{t-1}$ subsets containing this index
Compute each of these t-subset's expectation of separation when setting $*_i$ to s
Add to sum δ_s of all $\binom{k-1}{t-1}$ expectations for s
if some $\delta_s \geq density$ **then**
set $*_i$ to s
Subtract number of newly separated t-subsets from $unseparatedSubsets$
return A
end

Figure 4.2.1: Deterministic Construction of $ScHFs$ in Polynomial Time

Chapter 5

EMPIRICAL RESULTS

5.1 $ScHFs$ Constructed by Derandomized Algorithm

The algorithm presented in Figure 4.2.1 has been shown to deterministically construct $ScHFs$ in polynomial time. Due to the unexplored nature of these hash families, this algorithm will be used to generate the first general results, giving a baseline for the overall bounds of the size of these arrays with respect to the parameters. An important caveat to the runtime of this algorithm is that, while it is polynomial in terms of the size parameters, it is exponential in terms of the strength t. For this reason, the constructions will be limited to instances of moderately small $w \leq t \leq 6$ strengths. For the purposes of the analysis, however, existential bounds will suffice, and as such, the examined data points can be extended to larger strengths.

Figure 5.2.1 demonstrates the most noticeable property of $ScHFs$ when compared to $PHFs$: they exist on very small numbers of rows. The relaxation of the separation condition is such that any element can separate a row in any of the $\binom{k-1}{t-1}$ t-subsets in which it is present even when multiple instances of that symbol already exist. Even for multiplicity caps as low as $w = 2$, these families are extremely small. Using this simple construction method, for column sizes up to k =1,000,000,000, no more than 1,000 rows are required to ensure $ScHF$-separation up to $t = 6$, which is the highest calculable strength for which the expected separation calculation does not overflow a long integer in C/C++ during computation of $ScHFs$ with the specified parameters.

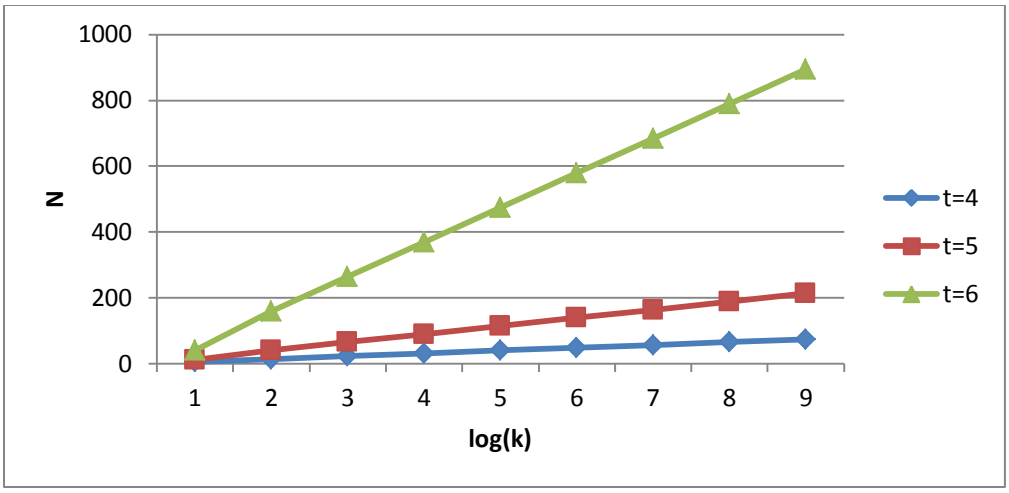

Figure 5.1.1: Rows of a $ScHF(N; k, 2, 3, t)$ for $k \leq 1{,}000{,}000{,}000$, $4 \leq t \leq 6$

In consideration of the application-specific analysis to follow, the results of this initial analysis are extremely positive. The derandomized construction was able to produce $ScHFs$ for all $k \leq$100,000,000 and $t \leq 6$ in under 800 rows, which as Table 5.2.1 demonstrates, is surpassed by PHF size for $k = 100$ on similarly restrictive parameter choices. The row values corresponding to $t > 6$ were extrapolated from the existence proof due to infeasibility of runtime. Appendix A has the remaining tables of results for all $w \geq 2$; however, Table 5.1.2 represents the tightest constraints on $ScHF$ separation and as such are the largest instances produced.

The impact of the size of these data structures suggests that due to the relaxation of the separation property by increasing the multiplicity cap, naïve construction techniques actually yield results that are far closer to minimality. In support of this concept as described in the following section, techniques such as post-optimization will be significantly less likely to succeed in significantly reducing the size of the solutions generated. Furthermore, the BES-specific implications of extremely small $ScHFs$ is explored in detail in Section 5.3.

$k\backslash t$	3	4	5	6	7	8	9	10	11	12
100	9	33	40	159	141	662	473	2610	1564	10043
1000	14	53	65	265	239	1139	825	4618	2803	18245
10000	19	72	90	370	335	1609	1172	6592	4021	26285
100000	24	92	115	475	432	2079	1519	8564	5236	34309
1000000	29	111	139	580	529	2549	1865	10536	6451	42332
10000000	34	131	164	684	625	3019	2212	12507	7666	50355
100000000	39	151	189	789	722	3488	2558	14478	8881	58377

Table 5.1.2: Current best known $ScHFN$ values for $w = 2$, $v = ceiling\left(\frac{t}{w}\right)$

5.2 $ScHFs$ Constructed by Post-Optimization

Don't-care post-optimization has been shown to be successful in measurably reducing the smallest known instances of CAs, $PHFs$, and other types of hash families [21]. By identifying elements of the hash family that are used in no unseparated t-subset, the algorithm slowly marks a row into don't care positions until it can be determined that no element in the row is used in any separation, and as such can be entirely discarded. Due to the large number of rows for a PHF of sufficient size and strength, after identifying a primary row, each subsequent row is expected to separate progressively less and less subsets when scanning for separations from this point. While it is not guaranteed that a given row can be eliminated, this algorithm employs a randomized local-minima escape strategy when too many failed iterations have occurred. What's more is that this algorithm can be stopped at any time and the output will still maintain its combinatoric properties, since the only modifications made are the removals of unnecessary elements.

A very notable property of $ScHFs$ is that, even for small values of w which is the closest a $ScHF$ can get to being a PHF without actually being one, the structures are extremely small when compared to $PHFs$. Since every $ScHF$ must have $v * w \geq t$, setting $w = 2$ and $v = ceiling\left(\frac{t}{w}\right)$ is placing the most constricting separation conditions possible on the family. Analyzing other choices of the parameters may produce different results; however, this choice was made to limit the scope of the potential variations to be studied to a set of restrictive conditions. Table 5.2.1 demonstrates the respective size of each structure under these conditions for $k = 100$. In these instances, post-optimizing a single row would represent between a much larger overall reduction in size compared to a PHF. Were this to happen, it would be a massive improvement in design, however, the fact that rows must be removed in integral steps combined with the very low number of rows causes this technique to fail to identify candidate rows for removal much more often.

T	$ScHFN$	$PHFN$
3	9	20
4	33	71
5	40	176
6	159	1087

Table 5.2.1: $ScHF$ ($w = 2$) and PHF Rows for $k = 100$ and Minimal v

It is important to note that the $ScHFs$ used for this comparison were not generated in an optimal fashion. These are all outputs of the derandomized construction method, which for other hash families, is demonstrably suboptimal. The small number of rows observed here becomes a crucial element in the performance of any post-optimization technique.

Due to the success with other forms of hashing, the don't-care post-optimization technique was applied to $ScHFs$ in an attempt to improve upon the minimality of the constructed $ScHFs$ produced by the derandomized algorithm. Despite the success of this algorithm elsewhere, the post-optimization of the $ScHFs$ generated using the techniques proposed by this thesis did not significantly reduce row count in the constructed instances. Initial analysis of these results suggest that this due to a property that can informally be referred to as "row weight". Specifically, this is the expected amount of separations that each row adds to the $ScHF$. Massive instances of $ScHFs$ exist on very few rows, and more importantly, despite the fact that many other separations have been made, the last row is still expected to cover a massive number of subsets. $ScHFs$ demonstrate a significantly higher row weight than $PHFs$, which could attribute to the observed behavior. When performing the don't-care post-optimization on these structures, the detection mechanism designed to avoid entering local minima triggered on every single execution. When disabling this feature of the algorithm, execution did not terminate after more than 36 hours of runtime on high multiplicity $ScHFs$.

As an additional performance metric for this post-optimization technique, the percentage of elements within a candidate row that were identified as don't-care positions were tracked. As an example, for $ScHF(N; \leq 100, 3, 4, 12)$, the highest percentage of a row to be identified as don't-care was approximately 26%. Greater success was found for families of multiplicity $w = 2$. These families are the closest in separation restriction to $PHFs$ and as such have the highest row counts. For $k \leq 1000$, $w = 2$, post-optimization reduced up to 12% of rows, but not in any predictably reliable fashion. Due to the intermittent results of this

approach, the $ScHF$ values analyzed for BES purposes represent the instances generated in Section 5.1. While the lack of success of optimization was not the desired outcome, Section 5.3 describes how the initial near-minimality of $ScHFs$ created by even naïve techniques are actually competitive in their performance as constructions for $BESs$

5.3 Analysis of Key Material and Broadcast Overheads

As previously described, for the purposes of analyzing the overheads associated with each scheme, all 1-resilient KPSs built utilize the same basic scheme. This decision, along with several other simplifying assumptions, was made to reduce the search space for constructing a BES. Despite this simplification, however, the veracity of the comparison remains; the parameters limited are ones that could potentially represent a $ScHF\ BES$ as performing worse than its ideal possible performance. Using a BES, the PHF parameters k and t are pre-defined by the BES constraints, and N is a function of k, v, and t. Thus, the only variable parameter to consider in terms of changing design overhead is v.

For a $ScHF\ BES$, k and t are also pre-defined, but there is both the number of symbols v to consider as well as the initial KPS strength that is being constructed and deployed. Restricting the number of symbols to $2 \leq v < k$ such that $v|k$ gives all $ScHF$ for a given k, t that can be constructed from a uniform ingredient KPS. Non-uniform ingredients massively increase the search space, and by restricting the consideration to a subset of possible $(k,t) - ScHF\ BESs$ and then selecting the best instance will only overestimate the minimal construction, not underestimate it. For every v, $2 \leq v < k$, the analysis considers all w, $2 \leq w < t$. This yields all possible starting KPS strengths except for those

for which would be constructing a PHF (and thus be trivially equal to the BES to which this scheme is being compared). Finally, since the ingredient KPS is itself a PHF-based scheme, the equality: $v_{PHF} = t_{PHF} = w$ is enforced. This is the most restrictive case for a PHF, and once again provides a pessimistic estimation that can only strengthen $ScHF$ claims made based on these results. If any $(k, t) - BES$ can be constructed more efficiently by using a pessimistic estimation for $ScHF$ overhead than by using a PHF's overhead, then the advantage over the previous scheme can still be claimed. In this analysis, efficiency is defined with respect to required number of broadcasts and average keys stored per user.

In addition to the message component broadcasts required by each scheme, there is a separate broadcast overhead that has not been considered. This overhead is the cost of broadcasting to all users the composition of the privileged subset P. It can be assumed that as subscribers join the BES, they are given a unique identifier, starting at 1 and increasing to k. Representing any subset $P \in \mathcal{P}(K)$ can be done in $|K|$ bits by transmitting a binary string in which a "1" represents a subscriber's presence in the privileged subset and a "0" represents his or her absence. Accordingly, all subscribers are able to identify P yet only the privileged members can compute all portions of the key.

The two broadcast overheads are distinguished from one another as the Set Identification Overhead (SIO) and the Broadcast Encryption Overhead (BEO), based on the function each overhead performs [1]. The Set Identification Overhead is absent from the compared overheads due to the fact that between schemes, the exact same identification must occur.

k:	3	4	5	6	7	8	9	10	11	12	13	14	15
$PHFN(k,3,3)$:	1	2	3	3	4	4	4	5	6	6	7	7	7
$PHFN(k,4,4)$:	-	1	5	5	8	8	8	11	12	14	15	16	17
$PHFN(k,5,5)$:	-	-	1	3	6	8	11	13	16	21	26	32	35

k:	16	17	18	19	20	21	22	23	24	25	26	27	28
$PHFN(k,3,3)$:	7	8	8	9	9	9	9	9	9	9	9	9	10
$PHFN(k,4,4)$:	19	20	22	23	24	26	27	28	29	30	31	32	33
$PHFN(k,5,5)$:	39	44	49	53	57	61	64	68	71	74	78	82	85

k:	29	30	31	32	33	34	35	36	37	38	39	40	41
$PHFN(k,3,3)$:	10	11	12	12	12	12	12	12	13	13	13	13	13
$PHFN(k,4,4)$:	34	35	36	37	38	39	40	40	41	41	41	41	41
$PHFN(k,5,5)$:	88	90	94	97	100	104	104	108	110	114	116	119	121

Table 5.3.1: Smallest known $PHFs$ of strength $t < 6$

Instrumental in this analysis is a compilation of minimal instances of $PHFs$ to use as ingredients for the final BES [31]. Not only are minimality constraints better known for smaller constructions, but they will also provide for a more accurate analysis of the overheads of the $ScHF$-based scheme.

In order to efficiently compare the overheads associated with both the PHF and the $ScHF$ scheme, the $PHFN$ values from Table 5.3.1 were written into a tool named Broadcast Encryption and Key Material Overhead ($BEKMO$) that rapidly generates broadcast encryption instances when provided with the appropriate

hash families as input. The comparative results of this section are based on the output of $BEKMO$.

An immediate finding in this comparison is the set of instances in which a $ScHF(1; k, v_1, w, t)$ out-performs a $PHF(N_2; k, v_2, t)$ in the construction of a $(k, t) - BES$. Consider a BES on 500 subscribers with a desired resilience of 6.

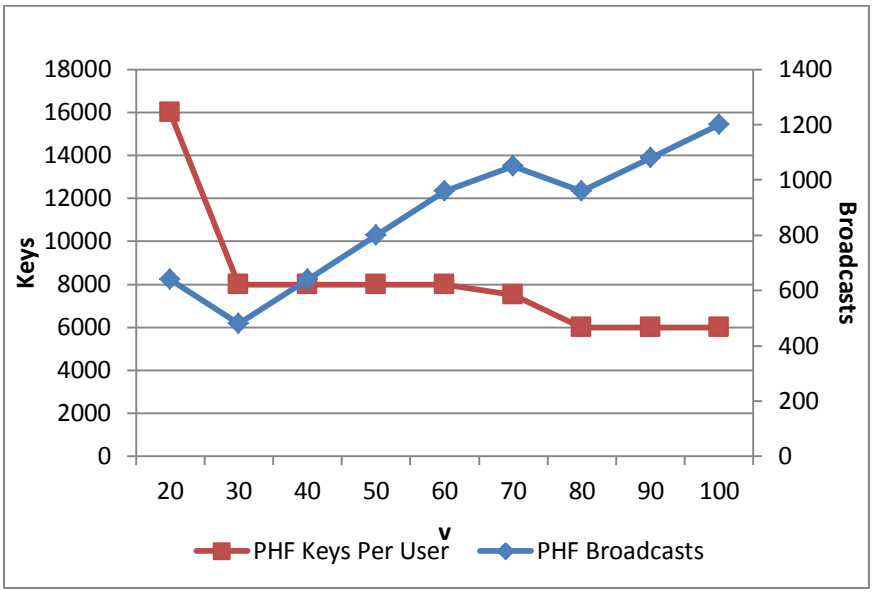

Figure 5.3.2: Broadcast and Key Material Overheads for $PHF\ (500,6) - BES$

Figure 5.3.2 charts the various options for selecting a PHF for the Fiat-Naor Scheme. While there is no specific metric forcing this decision, selecting a PHF close to the intersection of the two plots will give a moderate compromise between the two overheads. The output of the $PHFs$ used in this instance are the smallest known instances for their respective parameters. From this, an accurate comparison can be made. Table 5.3.3 gives the parameters associated with the best choice for each scheme built from instances generated by the derandomized construction. The solution in this situation is a trivial breakdown of

the scheme into various components. Due to utilizing a $ScHF$ on one row, it is not harnessing the inherent separation property across multiple $KPSs$ to great effect.

	Lowest Broadcast Overhead	Lowest Keys per User
PHF:	B = 480, KpU ≈ 8000	B = 960, KpU ≈ 6000
$ScHF$:	B = 156, KpU ≈ 4316	B = 540, KpU ≈1494

Table 5.3.3: Overhead Comparison for $(500,6) - BES$

In general, the information rate of PHF-based broadcast encryption is well known. Stinson demonstrated that following the Fiat-Naor construction, the scheme produces an information rate:

$$\rho = \frac{1}{N_{PHF}k}$$

indicating that in order to compute the single key associated with the privileged subset for a given broadcast, a user must obtain $\approx Nk$ times as many keys. Similarly, due to the recursive nature of the $ScHF\ BES$, the information rate for this scheme built from ingredient PHF_i is:

$$\frac{\rho_i}{N_{ScHF}v} = \frac{1}{N_{ScHF}v(N_{PHF_i} * \frac{k}{v})} = \frac{1}{N_{ScHF}N_{PHF_i}k}$$

when v is chosen minimally and so, it is clear that the information rate, and therefore the key material overhead for the $ScHF\ BES$ is dependent on the relationship between the size of the $ScHF$ solution and the size of its ingredients.

If, for any size instance, $N_{ScHF} * N_{PHF_i} < N_{PHF}$ holds, then the $ScHF$ provides a higher information rate to the subscribers. Using $BEKMO$ on schemes that were created by general methods that perform well asymptotically, a comparison between non-trivial instances of the schemes follows:

	$PHF(N, k, v, 5)$			Semi-Optimized $PHF(N, k, v, 5)$		
K	N	Info. Rate	Broadcasts	N	Info. Rate	Broadcasts
10	142	1/1420	710	14	1/140	70
20	247	1/4940	1235	57	1/1140	285
30	304	1/9120	1520	90	1/2700	450
40	343	1/13720	1715	119	1/4760	595

	$ScHF(N, k, v, 2, 5)$		
K	N	Info. Rate	Broadcasts
10	12	1/240	144
20	21	1/1260	378
30	26	1/3120	624
40	29	1/4640	696

Table 5.3.4: Comparison of $(k, 5) - BESs$ for PHF and $ScHF$ Based Schemes

When operating on minimal instances of $PHFs$, the performance of the PHF-based BES out-performs the $ScHF$ scheme, however, the problem with this comparison is that the larger the hash families, the less is known about their minimality in general. The $PHFs$ in this method were generated using the same derandomized technique that produced the $ScHFs$. Once a certain threshold of strength and size is passed, this has been shown to be the best general construction for $PHFs$. When comparing these schemes, it can be seen that $ScHFs$ constructed by the same method offer a moderate key material savings.

Despite this fact, the broadcast overhead tends to be comparable between the schemes as the size increases up to a certain threshold. As an additional factor of consideration, on very large $ScHFs$, partition sizes increase with $\frac{k}{v}$ and the ingredient $PHFs$ will begin to suffer from the same drop-off as the original PHF scheme. Should the $ScHF$ constructions improve as time goes on, these results will improve accordingly.

Scheme	Broadcast Encryption Overhead	Key Material Overhead
PHF	$O(\log(k))$	$O(k\ \log(k))$
$ScHF$	$O\left(\log(k)\log\left(\frac{k}{v}\right)\right)$	$O\left(k\log(k)\log\left(\frac{k}{v}\right)\right)$

Table 5.3.5: Overheads for PHF and $ScHF$ Broadcast Encryption Schemes

The comparison of both overheads between the two schemes is given in Table 5.3.5. The $ScHF$ scheme contains an additional logarithmic scaling, resulting from the overhead depending on both the $ScHF$ solution size as well as the ingredient PHF solution size. While these results provide the asymptotic behavior of each scheme, the empirical construction of the schemes demonstrates that the constant coefficient on size of the solutions is a non-trivial factor for small BES instances. Moreover, should v be defined as a function of k in both schemes, the additional scaling of the $ScHF$ scheme drops out entirely and they are equivalent asymptotically.

Further analysis into the overhead suggests that there exists a break-even point for which $ScHF$-based schemes with v independent of k begin to drop off due to their asymptotic behavior. As demonstrated in Section 5.1, easily

computable instances of $ScHFs$ are significantly smaller than $PHFs$ on equal columns and strength, but since the $BESs$ constructed from these $ScHFs$ scale with an additional $\log\left(\frac{k}{v}\right)$ factor, there is some size BES for which these schemes yield overheads greater than similarly generated PHF-based schemes. Creating schemes larger than this size favors other methods; however, should this point occur for BES parameters far exceeding practical demands, then the $ScHF$-based scheme offers a strict improvement. Determining this point requires a more in-depth analysis than the big-O asymptotes provided by this work. The constant factors affecting N_{PHF} and N_{ScHF} that are ignored by this analysis are necessary to ascertain the point of equivalence.

Chapter 6

CONCLUSION

By generalizing upon a well-known construction for broadcast encryption that inflates 1-resilient $KPSs$ into k-resiliency, this thesis has provided the fundamental rationale as well as the combinatorial basis for a new type of hash family. Scattering Hash Families generalize Perfect Hash Families and in order to analyze the properties of these families, techniques are formalized for their construction. Initial theoretic bounds have been given for these families as well, laying the foundation for more advanced approaches.

In practice, a simple deterministic construction provided excellent results for the construction of $ScHFs$. The method used to create this construction draws from a derandomization approach that creates strong instances of other types of hash families, however, when applied to $ScHFs$, the results appear to produce far smaller solutions for computable instances. Due in part to this near-minimality, the level of performance of post-optimization in practice does not carry over from other known related combinatorial structures.

While both the PHF-based BES and the $ScHF$-based BES have closely bound overhead behaviors, the $ScHF$-based scheme is able to exploit well-known minimal instances of $PHFs$ in situations in which the PHF-based BES is relying on a non-minimal solution. This allows the newly defined scheme to provide comparable, and in some situations, better performance. In addition to the explicit benefit analysis of key material and broadcast overheads, the $ScHF\ BES$ allows for an explicit, scalable design that can be efficiently constructed using simple deterministic methods.

Due to their novel nature, there is great potential for future work in the area of Scattering Hash Families. More efficient constructions that can handle larger strengths may allow for a better understanding of these structures in general. Additionally, the only security parameter considered in the scope of this thesis is the resilience of the scheme being deployed. Combining this parameter with features such as traitor-tracing and frame-proofing would strengthen the schemes in practice; however, determining the combinatorial requirements to obtain these properties was beyond the scope of this work. Such expansion has ultimately led to a wide application of modern broadcast encryption techniques, and applying these techniques to this new scheme might provide insight into $ScHF$ schemes or variants thereof.

REFERENCES

[1] A. Fiat and M. Naor, "Broadcast Encryption," *Lecture Notes in Computer Science,* vol. 773, pp. 480-491, 1994.

[2] M. Atici, S. S. Magliveras, D. R. Stinson and W. -D. Wei, "Some Recursive Constructions for Perfect Hash Families," *Journal of Combinatorial Designs,* vol. 4, pp. 353-363, 1996.

[3] C. J. Colbourn and J. H. Dinitz, "Perfect Hash Families," in *Handbook of Combinatorial Designs, Second Edition*, Chapman and Hall, 2007, pp. 556-568.

[4] S. Martirosyan and T. Trung, "Explicit constructions for perfect hash families," *Designs, Codes, and Cryptography,* vol. 46, no. 1, pp. 97-112, 2008.

[5] S. R. Blackburn, "Perfect hash families: probabilistic methods and explicit constructions," *J. Comb. Theory Ser. A,* vol. 92, no. 1, pp. 54-60, 2000.

[6] D. Deng, P. Li, G. v. Rees and Y. Zhang, "The Stein-Lovasz Theorem and Its Applications to Some Combinatorial Arrays," *Journal of Combinatorial Mathematics and Combinatorial Computing,* vol. 77, pp. 17-30, 2011.

[7] M. Ramkumar, "Broadcast Encryption Using Probabilistic Key Distribution and Applications," *Journal of Computers,* vol. 1, no. 3, 2006.

[8] J. L. Jin Hongxia, "Unifying Broadcast Encryption and Traitor Tracing for Content Protection," in *Annual Computer Scecurity Applications Conference*, Honolulu, 2009.

[9] M. Luby and J. Staddon, "Combinatorial Bounds for Broadcast Encryption," in *EUROCRYPT*, Espoo, 1998.

[10] P. D'Arco and D. R. Stinson, "Fault Tolerant and Distributed Broadcast Encryption," in *RSA*, San Francisco, 2003.

[11] M. Naor and B. Pinkas, "Threshold Traitor Tracing," in *CRYPTO*, Santa Barbara, 1998.

[12] E. Gafni, J. Staddon and Y. L. Yin, "Efficient Methods for Integrating Traceability and Broadcast Encryption," in *CRYPTO*, Santa Barbara, 1999.

[13] J. N. Staddon, D. R. Stinson and R. Wei, "Combinatorial properties of frameproof and traceability codes," Faculty of Mathematics, University of Waterloo, Waterloo, 2000.

[14] D. Stinson and P. Sarkar, "Frameproof and IPP Codes," in *Lecture Notes in Computer Science*, Berlin, Springer, 2001, pp. 117-126.

[15] D. R. Stinson, T. v. Trung and R. Wei, "Secure Frameproof Codes, Key Distribution Patterns, Group Testing Algorithms and Related Structures," *Journal of Statistical Planning and Inference,* vol. 86, no. 2, pp. 595-617, 1998.

[16] D. R. Stinson, "On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption," *Designs, Codes and Cryptography,* vol. 12, pp. 215-243, 1996.

[17] R. M. Solovay and V. Strassen, "A fast Monte-Carlo test for primality," *SIAM Journal on Computing,* vol. 6, no. 1, pp. 84-85, 1977.

[18] M. Rabin, "Probabilistic algorithm for testing primality," *Journal of Number Theory,* vol. 12, no. 1, pp. 128-138, 1980.

[19] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, 1983.

[20] J. H. Spencer, Ten Lectures on the Probabilistic Method, Montpelier: SIAM, 1994.

[21] P. Nayeri, C. J. Colbourn and G. Konjevod, "Randomized Postoptimization of Covering Arrays," in *Combinatorial Algorithms*, Berlin, Springer-Verlag, 2009, pp. 408-419.

[22] N. Alon and D. Moshkovitz, "Algorithmic construction of sets for k-restrictions," *ACM Transactions on Algorithms,* vol. 2, pp. 153-177, 2006.

[23] D. Naor, M. Naor and J. Lotspiech, "Revocation and Tracing Routines for Stateless Receivers," in *CRYPTO*, Santa Barbara, 2001.

[24] C. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal,* vol. 28, no. 4, pp. 656-715, 1949.

[25] S. K. Stein, "Two combinatorial covering theorems," *Journal of Combinatorial Theory,* vol. 16, no. 3, pp. 391-397, 1974.

[26] L. Lovász, "On the ratio of optimal integral and fractional covers.," *Discrete Math,* vol. 13, no. 4, pp. 383-390, 1975.

[27] D. S. Johnson, "Approximation Algorithms for combinatorial problems," *Journal of Computer and System Sciences,* vol. 9, pp. 256-278, 1974.

[28] R. C. Bryce and C. J. Colbourn, "The density algorithm for pairwise interaction testing," *Software Testing, Verificaton, and Reliability,* vol. 17, pp.

159-812, 2007.

[29] R. C. Bryce and C. J. Colbourn, "A density-based greedy algorithm for higher strength covering arrays," *Software Testing, Verification, and Reliability,* vol. 19, pp. 37-53, 2009.

[30] C. J. Colbourn, "Constructing Perfect Hash Families using a Greedy Algorithm," in *Coding and Cryptology*, Y. Li, S. Ling, H. Niederreiter, H. X. Wang, C. P. Xing and S. Y. Yang, Eds., World Scientific, pp. 109-118.

[31] R. A. Walker. II, "PHFtables," [Online]. Available: http://phftables.com. [Accessed 23 Jan 2011].

[32] M. Abdalla, Y. Shavitt and A. Wool, "Key management for restricted multicast using broadcast encryption," *Networking, IEEE/ACM Transactions on,* vol. 8, no. 4, pp. 443-454, 2000.

APPENDIX

$ScHFN$ FROM NON-CONSTRUCTIVE PROOF OF EXISTENCE

$w = 2$, $v = ceiling(t/w)$:

$k\backslash t$	3	4	5	6	7	8	9	10	11	12
100	9	33	40	159	141	662	473	2610	1564	10043
1000	14	53	65	265	239	1139	825	4618	2803	18245
10000	19	72	90	370	335	1609	1172	6592	4021	26285
100000	24	92	115	475	432	2079	1519	8564	5236	34309
1000000	29	111	139	580	529	2549	1865	10536	6451	42332
10000000	34	131	164	684	625	3019	2212	12507	7666	50355
100000000	39	151	189	789	722	3488	2558	14478	8881	58377

$w = 3$, $v = ceiling(t/w)$:

$k\backslash t$	3	4	5	6	7	8	9	10	11	12
100	1	8	19	56	36	88	317	131	354	1553
1000	1	12	31	94	61	151	553	231	634	2821
10000	1	17	43	130	86	214	786	330	909	4064
100000	1	21	54	167	111	276	1018	429	1184	5305
1000000	1	26	66	204	135	338	1251	527	1458	6546
10000000	1	30	78	241	160	401	1483	626	1733	7786
100000000	1	34	90	278	185	463	1715	725	2007	9026

$w = 4$, $v = ceiling(t/w)$:

$k\backslash t$	3	4	5	6	7	8	9	10	11	12
100	1	1	7	14	30	82	34	66	150	514
1000	1	1	11	23	51	140	60	116	269	933
10000	1	1	15	33	71	198	85	165	386	1343
100000	1	1	20	42	92	256	109	214	502	1753
1000000	1	1	24	51	112	313	134	264	618	2163
10000000	1	1	28	60	132	371	159	313	735	2573
100000000	1	1	32	69	153	429	184	362	851	2983

$w = 5$, $v = ceiling(t/w)$:

$k\backslash t$	3	4	5	6	7	8	9	10
100	1	1	1	7	12	21	42	108
1000	1	1	1	11	20	36	73	191
10000	1	1	1	15	27	51	104	273
100000	1	1	1	19	35	66	135	355
1000000	1	1	1	23	43	81	165	436
10000000	1	1	1	27	51	96	196	518
100000000	1	1	1	30	58	111	226	599