Linear-Time Encodable Codes and Cryptography

Erez Druk

Technion - Computer Science Department - M.Sc. Thesis MSC-2013-22 - 2013

Linear-Time Encodable Codes and Cryptography

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Erez Druk

Submitted to the Senate of
the Technion — Israel Institute of Technology
Tishrei 5774 Haifa September 2013

The research thesis was done under the supervision of Prof. Yuval Ishai in the Computer Science Department.

I would like to thank my advisor, Prof. Yuval Ishai, for teaching me how research is done, for providing most of the major ideas, insights, questions and answers related to this work and for always having infinite patience and time for me. I would also like to thank Ariel Gabizon, Elad Haramaty, Eyal Kushilevitz, Netanel Raviv, Ronny Roth and Amir Yehudayoff for their helpful input. I would like to thank Yardena Kolet for her ability to make everything possible. Lastly, I would like to thank my family for always being there for me.

The generous financial support of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract

An error-correcting code with minimal distance d encodes a k-bit message into an n-bit codeword such that any two distinct codewords differ in at least d coordinates. It is well known that a random code, or even a random linear code, has a good minimal distance with high probability. Moreover, the conjectured intractability of decoding random linear codes has recently found several applications in cryptography.

A major disadvantage of random linear codes is that their encoding complexity grows quadratically with the message length. Motivated by this disadvantage, we present a new randomized construction of linear error-correcting codes which can be encoded in linear-time and yet enjoy several useful features of random linear codes. Our construction is based on a linear-time computable hash function due to Ishai, Kushilevitz, Ostrovsky and Sahai [25].

We demonstrate the usefulness of our new codes by presenting several applications in coding theory and cryptography. These include the first family of linear-time encodable codes whose minimal distance meets the Gilbert-Varshamov bound, the first nontrivial linear-time secret sharing schemes, and plausible candidates for symmetric encryption and identification schemes which can be conjectured to achieve better asymptotic efficiency/security tradeoffs than all current candidates.

Abbreviations and Notations

q	—	Field size
$\mathbb{F}_q$	—	Galois field GF(q)
C	—	Error correcting code
$C^\perp$	—	Dual code
$\mathcal{C}$	—	Code family
k	—	Message length
$k^\perp$	—	Dual code message length
n	—	Word length
R	—	Rate
$R^\perp$	—	Dual rate
d	—	Code's minimal distance
$d^\perp$	—	Dual distance
δ	—	Code's relative distance
$\delta^\perp$	—	Dual relative distance
$\mathbb{N}$	—	The set of natural numbers
$[n]$	—	The set {1,2,. . . ,n}
A, B, C	—	Matrices
x, y, z	—	Vectors
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	—	Distributions
$a \in_R \mathcal{A}$	—	Random choice of a according to A
$\Delta(\cdot, \cdot)$	—	Hamming distance function
$.^T$	—	Transposition function
$H_q(\cdot)$	—	q-ary entropy function
$H_q^{-1}(\cdot)$	—	Minimal q-ary entropy inverse function
GV	—	Gilbert-Varshamov

Chapter 1

Introduction

This work is concerned with error correcting codes and their applications in cryptography. We start by explaining some of the standard coding theory terminology we will use.

A *linear code* over a finite field $\mathbb{F}$ with message length k and block length n is a k-dimensional linear subspace $C \subseteq \mathbb{F}^n$. The *rate* of C is the normalized dimension $R(C) = k/n$. The *minimal distance* of C, denoted by d, is the minimal Hamming distance between any two distinct codewords $c, c' \in C$, namely the number of entries in which c and c' differ. The *relative distance* of C is the normalized minimal distance $\delta(C) = d/n$. Using an injective function $E : \mathbb{F}^k \to C$, called an *encoder*, any message $x \in \mathbb{F}^k$ can be encoded into a codeword $c = E(x)$ such that even if up to d entries of c are erased or up to $d/2$ entries are modified, there is still sufficient information to recover x. An algorithm which recovers x from a corrupted version of $E(x)$ is called a *decoder*. All of the above notions can be naturally generalized to an arbitrary (rather than linear) code $C \subseteq \Sigma^n$ over the alphabet Σ.

The minimal distance and rate of C are the two major parameters by which the quality of C as an error correcting code is assessed. Increasing any of these values increases the quality of the code in hand either by improving its error handling capabilities or by increasing the amount of information carried by the codewords. Unfortunately, these code parameters are contradictory in nature and the possible tradeoffs between them are yet to be fully understood. One of the early goals of coding theory was to construct *asymptotically good codes,* namely infinite families of codes in which both the rate and relative distance are bounded away from zero.

As often is the case with combinatorial objects, a *random* family of codes, whether linear or not, is asymptotically good with high probability (cf. [19]). In fact, random codes achieve the best known rate/distance tradeoff in the binary case and are only slightly inferior to the best known constructions over larger fields (achieved by codes based on algebraic geometry [40]). The rate/distance tradeoff achieved by random codes is captured by the *Gilbert-Varshamov (GV) bound* [15, 41]. The GV bound tells us that for any prime power q and $\delta \in (0, 1-1/q)$ there is an infinite family C_k of linear codes over $\mathbb{F}_q$ with relative minimal distance at least δ and rate $R(C_k)$ which tends to $1 - H_q(\delta)$, where $\mathrm{H}_q(x) = x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x)$ is the q-ary entropy function. In particular, the GV bound shows the existence of asymptotically good codes.

In addition to maximizing the rate and minimal distance of codes, another important optimization goal is to minimize the *computational complexity* of encoding and decoding. By default, we will measure computational complexity by the size of an arithmetic circuit performing the computation (whose gates may perform addition, subtraction or multiplication of two field elements). We will also refer to this measure of complexity as the encoding and decoding *time*. See Section 2.2 for more details on the computational model.

The main focus of this work is on minimizing the encoding complexity of codes. Random codes fail miserably by this measure, having exponential encoding complexity in the general case and quadratic in the linear case. Since the best asymptotic encoding complexity one could hope for is linear in n, it is natural to ask whether there are asymptotically good families of linear-time encodable codes. The first proof that such codes exist is due to Gelfand, Dobrushin and Pinsker [34], who presented a randomized construction of linear-time encodable linear codes over the binary field which have positive rate and relative minimal distance. An explicit construction of such codes, which also admits a linear-time[1] decoding algorithm, was given in a celebrated work of Spielman [39].

The concrete rate/distance tradeoff achieved by Spielman's codes is far from the GV bound. Guruswami and Indyk [23] construct linear-time en-

[1] Linear-time decoding can be carried out in a more liberal computation model, namely a RAM model. In our default circuit model the codes of [39] and similar "linear-time decodable" codes can be decoded in quasi-linear-time.

codable codes whose rate and distance parameters can get arbitrarily close to the GV bound. Unfortunately, the closer one wishes to get to the bound the larger the size of the underlying field becomes. These results leave open the existence of linear-time encodable codes which meet the GV bound, or even get close to this bound in the binary case. This question serves as a primary motivating question for our work.

1.1 Our Contribution

We present a randomized construction of linear-time encodable codes which enjoy several useful properties of random linear codes. In particular, our construction answers the above question in the affirmative over every finite field, giving the first families of linear-time encodable codes which meet the GV bound.

Our construction relies on a pseudorandom object we refer to as a *linear uniform output family*. A linear uniform output family over a finite field $\mathbb{F}$ is a family of linear functions $A : \mathbb{F}^k \to \mathbb{F}^n$ such that for any fixed nonzero $x \in \mathbb{F}^k$, a random choice of A from the family makes $A(x)$ uniformly distributed over $\mathbb{F}^n$. We apply a variant of a construction of linear-time computable pairwise independent hash functions due to Ishai, Kushilevitz, Ostrovsky and Sahai [25] to obtain a linear uniform output family in which each A is computable in linear time. Furthermore, $A(x)$ can be computed in time $O(n)$ given x and a uniformly random succinct description of A of length $O(n)$. This construction applies over any finite field $\mathbb{F}$ and for any choice of message length k and block length n.

A similar construction of linear uniform output family, over the binary field, was recently used by Baron, Ishai, and Ostrovsky [5] for obtaining linear-time computable hardcore functions and pseudorandom generators. Our construction is somewhat simpler and applies over general fields. In contrast to our work, the applications in [5] rely on families which shrink (rather than expand) the input, and require the family to satisfy an additional "bilinearity" property.

The construction of linear-time computable linear uniform output families is presented in Chapter 3, which also presents applications in coding theory. In particular, Chapter 3 presents the application to linear-time encodable codes meeting the GV bound, and also establishes other properties

of these codes which are used in Chapters 4 and 5.

In contrast to the linear-time encodable codes from [39, 23], our codes are not fully[2] explicit. However, even without the linear-time encoding requirement, the question of explicitly constructing codes which meet the GV bound is wide open (cf. [22]).

Another difference between our codes and the linear-time encodable codes from [39, 23] is that we do not know whether our codes support efficient decoding. This is not necessarily a problem for two reasons. First, there are natural applications of codes that only require decoding in the presence of *erasures* rather than errors, which can be done in polynomial time for any linear code. Second, and perhaps more interestingly, if the decoding problem for our codes indeed turns out to be intractable (as we conjecture), this can potentially make them useful for cryptographic applications. Indeed, there are quite a few cryptographic constructions that rely on the conjectured intractability of decoding random linear codes. Our new family serves as a plausible candidate for a smaller family of linear codes that has linear description and encoding complexity and yet enjoys the minimal distance and intractability features of random linear codes.

In Chapters 4 and 5 we discuss applications of our codes in the field of cryptography. We make a distinction between applications to "information-theoretic cryptography," which require security against computationally unbounded adversaries but whose security can be proved unconditionally, and applications to "complexity-based cryptography," which settle for security against bounded adversaries but need to rely on unproven intractability assumptions. These two types of applications differ not only in the security guarantees they provide and the type of assumptions they rely on, but also in the required properties of the underlying codes.

We begin in Chapter 4 with an information-theoretic application, obtaining the first linear-time implementation of nontrivial secret sharing schemes. More concretely, we show that for any constants $0 < c_s < c_r < 1$ there is a "near-threshold" scheme for sharing a secret bit among n players, such that any set of at most $c_s n$ players learn nothing about the secret, any set of at least $c_r n$ players can jointly reconstruct the secret, and the share generation

[2]Our codes are semi-explicit in the sense that a linear-size encoding circuit meeting the required bound on the minimal distance can be generated in probabilistic polynomial time with negligible failure probability.

can be implemented in time $O(n)$ given the secret and uniformly random bits. This should be compared to the well known secret sharing scheme due to Shamir [37], which achieves a sharp threshold between the secrecy and reconstruction requirements, but its implementation requires $n \cdot \text{polylog}(n)$ time. This application relies on known connections between linear codes and secret sharing, and exploits the fact that codes from our family (like random linear codes) do not only have a good minimal distance but also a good *dual distance*.

For the complexity-based cryptographic applications, discussed in Chapter 5, we draw inspiration from the well studied problem of learning parity with noise (LPN) and its growing number of cryptographic applications. In the LPN problem one is asked to recover a random k-bit vector x using samples of the form $(A, Ax+e)$, where A is a random $n \times k$ binary matrix and e is a random n-bit noise vector of a low Hamming weight. The intractability of the LPN problem is a question with a "win-win" flavor: If the problem is indeed intractable, then a variety of efficient cryptographic constructions of objects such as pseudorandom generators [18, 7], symmetric encryption schemes [16, 4], identification schemes [24] and more, are secure. On the other hand, if the problem turns out to be tractable (even in a weak sense), this would be considered a major breakthrough in coding theory, showing that random linear codes support efficient decoding.

Most of the applications of LPN in cryptography rely (either explicitly or implicitly) on the fact that the LPN problem admits a search to decision reduction. Roughly speaking, this means that recovering x from $(A, Ax+e)$ is equivalent to distinguishing between $(A, Ax+e)$ and (A, U_n), where U_n is a uniformly random n-bit vector. We start our discussion of applications in complexity-based cryptography by showing a similar search to decision reduction for our new family of codes. This allows us to make the more conservative hardness assumption that it is hard to solve the search (i.e., decoding) problem, while using the stronger hardness-of-decision assumption in the applications. We then observe that LPN-based constructions from the literature can be applied with arbitrary distributions of the matrix A as long as the generalized intractability assumption is met.

As applications, we get plausible candidates for stateless symmetric encryption and identification schemes which may achieve better asymptotic efficiency/security tradeoffs than all current candidates. Concretely,

if the time complexity of the honest parties is n, then the best known attacks [29, 27] require $2^{O(n/\log\log n)}$ time. This should be compared with the original LPN-based constructions, which require the honest parties to spend $O(n^2)$ time for achieving a similar security guarantee, and an optimized variant suggested in [16] which requires $n \cdot \text{polylog}(n)$ time. Similarly to LPN-based constructions, our constructions have a win-win flavor in that disproving their conjectured security would lead to interesting progress in coding theory.

Chapter 2

Preliminaries

2.1 Basic Notation

For a prime power q, we denote by $\mathbb{F}_q$ the field $\mathrm{GF}(q)$, i.e., the finite field with q elements. We let $\mathbb{N}$ denote the set of natural numbers. For any $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. We use lower case letters i, j, k to denote indices, x, y, z to denote column vectors in $\mathbb{F}_q^n$, and upper case letters A, B, C to denote matrices in $\mathbb{F}_q^{n \times k}$. The transpose of a column vector x or a matrix A are denoted by x^T and A^T respectively. We let x_i denote the i-th entry of x and a_{ij} denote the entry in the i-th row and j-th column of A.

2.2 Models of Computation

The main model of computation by which we measure the efficiency of algorithms in this work is the model of arithmetic circuits over a finite field $\mathbb{F}$. Each wire in one of these circuits carries a field element. Each gate in the circuit has two input wires, and outputs the addition, subtraction, or multiplication of its two inputs. We allow gates to have an arbitrary fan-out and a fan-in of either 2 or 0, where a gate with fan-in 0 is labeled by either an input variable or a constant. We also consider *probabilistic circuits* which contain randomness gates with fan-in 0. The values of such gates are picked uniformly and independently from $\mathbb{F}$. The *size* and *depth* of a circuit are the number of wires and the length of the longest path in the graph

of the circuit, respectively. See [9] and [36] for more precise definitions of this circuit model. We will consider polynomial-time uniform families of circuits, namely an infinite family of circuits C_k over $\mathbb{F}$, where C_k has k inputs and there is a polynomial-time algorithm which on input 1^k outputs a description of C_k.

Previous works on linear-time encodable and decodable codes (e.g., [38, 39]) also consider the model of a RAM machine [1]. A RAM has a central processor that can access data from a memory. This processor should have a few basic operations: addition, subtraction, read from memory, store to memory, and branch if zero. Each of these operations costs one unit of time. While the circuit model is generally considered more restrictive, the two models are technically incomparable. However, allowing a polynomial-time computable advice that depends only on the input length, any uniform family of circuits of size $s(k)$ can be simulated by such a RAM machine running in time $O(s(k))$.

We will often argue that some object X can be computed in linear-time or by linear-size circuits. By this we mean that the functions associated with X can be computed by a polynomial-time uniform family of linear-size circuits, implying that they can also be computed by a linear-time RAM machine with polynomial-time computable advice. For the sake of concreteness, our default model will be that of uniform arithmetic circuits.

2.3 Coding Theory

A *code* over an *alphabet* Σ with *block length* n is a subset $C \subseteq \Sigma^n$. For two vectors $u, v \in \Sigma^n$, we denote by $\Delta(u, v)$ the *Hamming distance* between u and v, that is, the number of coordinates in which u and v differ. The *Hamming weight* of y is $\Delta(y) = \Delta(y, \vec{0})$. The code's *minimal distance* is defined to be $\min_{c,c' \in C, c \neq c'} \Delta(c, c')$ and is denoted by d while the code's *relative distance* is the value $\delta(C) = \frac{d}{n}$.

The code's *message length* and *rate* are defined as $k = \log_q |C|$ and $R = \frac{k}{n}$, respectively, where $q = |\Sigma|$. We often view the code as the image of an injective *encoding function* (or *encoder*) $E_C : \Sigma^k \to \Sigma^n$. A code C as above is referred to as an $(n, k, d)_q$ code. In our case, Σ will usually be a finite field $\mathbb{F} = \mathbb{F}_q$ and C a k-dimensional linear subspace of $\mathbb{F}^n$. In such a case we say that C is a *linear code* or an $[n, k, d]_q$ code and refer to the orthogonal

space of C as the *dual code* whose minimal distance $d^{\perp}$ is referred to as the *dual distance* of C. An $[n, k, d]_q$ code C admits an encoding function of the form $E_C(x) = Ax$, where A is a $n \times k$ matrix of rank k over $\mathbb{F}$ referred to as a *generator matrix* for C.

We define analogues of the quantities above for infinite families of codes. An infinite family of codes is a family $\mathcal{C} = \{C_i | i \in \mathbb{N}\}$ where C_i is a code of block length n_i with $n_i > n_{i-1}$ and message length k_i. The rate of $\mathcal{C}$ is defined to be $R(\mathcal{C}) = \liminf_i R(C_i)$ and the relative distance is $\delta(\mathcal{C}) = \liminf_i \delta(C_i)$. We say that $\mathcal{C}$ is *asymptotically good* if $R(\mathcal{C}), \delta(\mathcal{C}) > 0$. We say that $\mathcal{C}$ is encodable in time $t(n)$ if each C_i is the image of an injective encoding function $E_i : \mathbb{F}_q^{k_i} \to \mathbb{F}_q^{n_i}$ where E_i is computable in time $t(n_i)$. When $\mathcal{C}$ is a family of linear codes we require each E_i to be a linear function.

The following definitions and claims are taken almost verbatim from [19, Notes 2].

Definition 2.3.1 *For integers q, n, l denote by $\mathrm{VOL}_q(n, l)$ the number of elements in a Hamming ball of radius l in $\mathbb{F}_q^n$. This value is independent of the center of the ball and equals*

$$\mathrm{VOL}_q(n, l) = \sum_{i=0}^{l} \binom{n}{i} (q-1)^i$$

Definition 2.3.2 *For a positive integer $q \geq 2$,* the q-ary entropy function $\mathrm{H}_q : [0, 1] \to \mathbb{R}$ *is defined as follows:*

$$\mathrm{H}_q(x) = x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x)$$

For every $y \in [0, 1]$ there exists a unique value $x \in [0, 1 - 1/q]$ such that $H_q(x) = y$. This allows us to define a inverse entropy function in following manner.

Definition 2.3.3 *For a positive integer $q \geq 2$, the* inverse q-ary entropy function $\mathrm{H}_q^{-1} : [0, 1] \to [0, 1]$ *is defined as $\mathrm{H}_q^{-1}(y) = x$ where $x \in [0, 1 - 1/q]$ is the unique value satisfying $H_q(x) = y$.*

Lemma 2.3.4 *For an integer $q \geq 2$ and $p \in [0, 1 - \frac{1}{q}]$,*

$$\mathrm{VOL}_q(n, pn) \leq q^{\mathrm{H}_q(p)n}$$

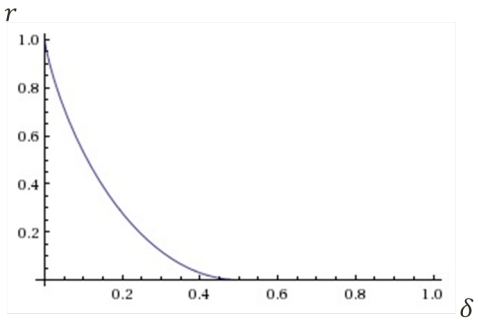

Figure 2.1: Gilbert-Varshamov bound over the binary field

Theorem 2.3.5 (Asymptotic Gilbert-Varshamov bound) *For every prime power q and $\delta \in [0, 1 - \frac{1}{q}]$, there exists an infinite family $\mathcal{C}$ of linear codes over $\mathbb{F}_q$ with relative distance $\delta(\mathcal{C}) \geq \delta$ and rate $R(\mathcal{C}) \geq 1 - \mathrm{H}_q(\delta)$.*

Figure 2.1 shows a graphical illustration of the Gilbert-Varshamov bound over the binary field.

2.4 Probability

We use calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$ to denote discrete probability distributions. We use the notation $a \in_R \mathcal{A}$ to denote a random choice of a from the distribution $\mathcal{A}$. For $n \in \mathbb{N}$ and $\mu \in [0, 1]$, we denote by $\textsc{Ber}_\mu^n$ the distribution over n-bit vectors where each entry is chosen to be 1 with probability μ (independently of the other entries). The uniform distribution over n bit vectors, $\textsc{Ber}_{1/2}^n$, is also denoted by U_n. For a sequence of (possibly randomized) operations $O_1, \ldots, O_m$ and an event A, we let $\Pr[O_1; \ldots; O_m : A]$ denote the probability of the occurrence of A after the sequential execution of $O_1, \ldots, O_m$.

Lemma 2.4.1 (XOR-lemma [13]) *Let $X = (X_1, \ldots, X_n)$ be a random variable distributed over $\mathbb{F}_q^n$. Then X is uniformly distributed if and only if for every nonzero $y \in \mathbb{F}_q^n$, the random variable $y^T X$ is uniformly distributed over $\mathbb{F}_q$.*

Definition 2.4.2 (Extractor for Bit-Fixing Sources) *An* (l, m, n) extractor for bit-fixing sources *over* $\mathbb{F}$ *is a function* $g : \mathbb{F}^m \to \mathbb{F}^n$ *such that for every distribution* X *over* $\mathbb{F}^m$ *that fixes at most* l *entries and distributes the rest uniformly, the output* $g(X)$ *is uniformly distributed over* $\mathbb{F}^n$*.*

Definition 2.4.3 (Pairwise independence) *A distribution* $\mathcal{H}$ *over functions* $h : X \to Y$ *is* pairwise independent *if for every* $x, x' \in X$ *such that* $x \neq x'$ *and every* $y, y' \in Y$ *it holds that*

$$\Pr_{h \in_R \mathcal{H}}[h(x) = y \wedge h(x') = y'] = |Y|^{-2}.$$

When $\mathcal{H}$ *is a set or multiset of functions, the corresponding distribution is understood to be uniform over* $\mathcal{H}$*.*

2.5 Asymptotics

A function $\epsilon(k)$ is said to be *negligible* if for every polynomial p and all sufficiently large k we have $\epsilon(k) < 1/p(k)$. When we say that an infinite sequence of events E_k occurs with negligible probability, we mean that the probability $\epsilon(k)$ of E_k is negligible. We will make a standard use of big-O and big-Ω notation.

In asymptotic expressions that involve multiple variables, the variables not explicitly included inside the $O(\cdot)$ or $\Omega(\cdot)$ are treated as constants. For instance, the expression $\epsilon(q, n) = q^{-\Omega(n)}$ means that for every q there exists $\theta > 0$ such that for all sufficiently large n, $\epsilon(q, n) \leq q^{-\theta n}$.

Chapter 3

Linear-Time Encodable Codes Meeting the Gilbert-Varshamov Bound

In this chapter we present a probabilistic construction of linear-time encodable codes that meet the GV bound. Our main tool is a new pseudorandom object which we call a *linear uniform output family*. Section 3.1 presents a linear-time computable linear uniform output family. In Section 3.2 we discuss applications to coding theory, including linear-time encodable codes that meet the GV bound.

3.1 Linear Uniform Output Families

Definition 3.1.1 (Linear Uniform Output Family) *A* (q, k, n) linear uniform output family *is a distribution* $\mathcal{A}$ *over linear functions* $A : \mathbb{F}_q^k \to \mathbb{F}_q^n$*, such that for every nonzero* $x \in \mathbb{F}_q^k$ *and* $y \in \mathbb{F}_q^n$ *it holds that* $\Pr_{A \in_R \mathcal{A}}[A(x) = y] = q^{-n}$*. Equivalently, for every fixed nonzero* $x \in \mathbb{F}_q^k$ *the distribution of* $A(x)$ *induced by a choice of* A *according to* $\mathcal{A}$ *is uniform over* $\mathbb{F}_q^n$*.*

For notational convenience, we will identify a function A in the support of $\mathcal{A}$ with the $n \times k$ matrix representing it.

A well known linear uniform output family is the family of $n \times k$ Toeplitz matrices over $\mathbb{F}_q$. Similarly to our construction, each Toeplitz matrix A has

a linear size description. However, the function corresponding to A is only known to be computable in quasi-linear-time (see [30, 28]).

We now present a linear-time computable linear uniform output family. Our construction is a variant of the pairwise independent hash function family from [25, Theorem 3.3]. For the sake of completeness, we start with a high level overview of the latter.

Overview of the construction from [25] The key idea of the construction is to use a linear number of constant-size pairwise independent hash functions to implement a large pairwise independent hash function. This is done in three phases. First, a linear-time encodable code C is being used to encode the input x, resulting in an encoding y. For the construction as a whole to work, the alphabet size may be increased at this phase to allow the fractional distance to be bigger than $1/2$. Next, an independent constant-size pairwise independent hash function h_i is applied to each i'th symbol of y, resulting in a randomized encoding y'. Finally, a linear-time computable extractor for bit-fixing sources, g, is applied to y' obtaining the final output. Such an extractor can be obtained by transposing a linear-time encoding function of a linear code with good minimal distance, where the transpose mapping can be shown to also be implementable in linear time. Each function in the family is hence of the form $g \circ \vec{h} \circ E_C$ and is specified by the choice of $\vec{h} = (h_i)$. Since all three phases may be implemented in linear time, so does the entire construction.

An informal argument as to why the above function is pairwise independent is as follows. For every distinct inputs x_1, x_2 we have that the corresponding encodings y_1 and y_2 differ in many of the coordinates and hence y'_1 and y'_2 are distributed uniformly and independently in all these coordinates. In order to guarantee that both outputs are completely uniform and independent we shrink them using an extractor for bit fixing sources which "ignores" those places where y'_1 and y'_2 are the same, turning partial independence into full independence.

To construct a linear uniform output family, we instantiate the above construction so that all building blocks are implemented by linear functions. Since we will be interested not only in the binary case, we present a more general form of the original building blocks that applies over any finite field.

The construction relies on asymptotically good families of linear-time

encodable linear codes. Spielman [39] obtains an explicit construction of such codes over the binary field. Here we generalize this construction and its analysis to apply over any finite field. We rely on the explicit lossless expanders of Capalbo et al. [11, Theorem 7.1].

Lemma 3.1.2 *There exists a constant $c_0 > 0$ such that for every positive integers n, t such that $t \leq n$ and every $\epsilon > 0$ there exists a bipartite graph $G = (L, R, E)$ with $|L| = n, |R| = m = \lfloor n/t \rfloor$ and the following properties:*

- **Degree.** *The degree of every $a \in L$ is $\Delta = \mathrm{poly}(\log t, 1/\epsilon)$.*
- **Expansion.** *For every set $X \subset L$ with $|X| \leq \frac{c_0 \epsilon m}{\Delta}$, if $Y = \Gamma(X)$ is the set of neighbors of X in G, then $|Y| \geq (1 - \epsilon)\Delta|X|$.*
- **Explicitness.** *G can be constructed in $\mathrm{poly}(n, 2^{\mathrm{poly}(t,1/\epsilon)})$ time.*

Lemma 3.1.3 *There exist constants $d_1 > 1$ and $d_2 > 0$ such that for every field $\mathbb{F}$ there exists a family of $[\lfloor d_1 k \rfloor, k, \lceil d_2 k \rceil]$ linear codes over $\mathbb{F}$ which can be encoded by a uniform family of linear-size arithmetic circuits consisting of addition and fan-out gates only.*

Proof: The construction is a simplified version of Spielman's code [39] generalized to any finite field. The only difference is that we replace XOR with general addition gates over $\mathbb{F}$, the base code of the recursive construction is over $\mathbb{F}$ as well and since we only care about the minimal distance of the code the construction involves lossless expanders instead of error reducing codes which are required for the decoding process of the original code.

We will now describe a recursive construction of a family of encoding functions $E_h : \mathbb{F}^{n_0 2^{h-2}} \to \mathbb{F}^{n_0 2^h}$ for some constant n_0 divisible by 4 such that a circuit of size $O(2^h)$ which computes E_h using addition gates can be generated in time $\mathrm{poly}(2^h)$ and the code defined by E_h has constant relative distance. The general case follows by padding an input of length k with zeros such that the obtained input length is the closest value of the form $n_0 2^h$. Doing so affects the rate by a constant factor and preserves the relative distance of the code encoded by E_h.

The base encoding function E_0 would be the encoding function of a linear code C_0 of block length n_0, rate $1/4$ and relative distance $\delta' > 0$. We

set $\epsilon < 0.1$ and for $h \geq -1$ we denote by $G_h = (L_h, R_h, E'_h)$ the expander graph from Lemma 3.1.2 with $|L_h| = n_0 2^h, |R_h| = n_0 2^{h-1}, c_0 > 0$ and $\Delta = \text{poly}(1/\epsilon)$. We assume that Δ is not too small (say $\Delta > 10$; Since Δ is a function of $1/\epsilon$ we do not lose generality).

For every $h > 0$ we define E_h to encode a message m of length $n_0 2^{h-2}$ in the following manner. First, G_{h-2} is used to compute a message m_1 of length $n_0 2^{h-3}$. More concretely, $(m_1)_j = \Sigma_{i \in \Gamma(\{j\})} m_i$ is the sum of the entries of m indexed by the neighbors of $j \in R_{h-2}$. We next compute $c_1 = E_{h-1}(m_1)$ whose length is $n_0 2^{h-1}$. We use G_{h-1} and c_1 as done above to compute c_2 whose length is $n_0 2^{h-2}$. The final output is $c = m \circ c_1 \circ c_2$ whose length is $n_0 2^h$.

We will now show by induction that for $\delta = \min\{\frac{c_0 \epsilon}{8\Delta}, \delta'\}$ every encoding function E_h defines a code C_h with relative distance least δ. The other assertions on E_h are easy to validate. Let m be a nonzero message of length $n_0 2^{h-2}$ and consider the following two possible cases. First, when the relative Hamming weight of m is at least 4δ we have that the relative weight of $c = m \circ c_1 \circ c_2$ is at least δ. Otherwise, let X denote the index set of the nonzero entries of m. Since the relative weight of m is at most $4\delta \leq \frac{c_0 \epsilon}{2\Delta}$ we have that $|X| \leq \frac{c_0 \epsilon |B_{h-2}|}{\Delta}$ and hence $\Gamma(X) \geq (1-\epsilon)\Delta|X|$. A simple counting argument shows that at least $(1-2\epsilon)\Delta|X| > 8|X| > 0$ of the vertices in B_{h-2} have a unique neighbor in X. All these vertices correspond to a nonzero entry in m_1. Since m_1 is nonzero, the relative weight of c_1 is at least $\delta' \geq \delta$. We have two possible sub-cases to consider. If the relative weight of c_1 is at least 2δ then we are done. Otherwise, the relative weight of c_1 is at least δ and at most 2δ. An argument as above with X' denoting the index set of the nonzero entries of c_1 shows the the weight of c_2 is at least $(1-2\epsilon)\Delta|X'| \geq 8\delta n_0 2^{h-1} > \delta n_0 2^h$ as promised. ■

The first component of the construction, namely the code C in the overview, is obtained by permuting and grouping symbols of a code as above into constant-size blocks so that its relative minimal distance becomes a sufficiently large constant. The construction of C is similar in spirit to the expander based construction of Guruswami and Indyk [21] which applies the distance boosting technique of ABNNR [3] to an asymptotically good linear-time encodable code. We will need the following object which is a simplified version of the expander graph used in [20, Theorem 11.4].

Lemma 3.1.4 *There exists a constant $c_1 > 0$ such that for every $\epsilon > 0$ and positive integer n there exists a bipartite graph $G = (L, R, E)$ with $|L| = |R| = n$ and the following properties:*

- **Degree.** *The degree of every vertex in G is $\Delta = \lceil c_1/\epsilon^3 \rceil$.*
- **Expansion.** *For every set $X \subset L$ with $|X| \geq \epsilon n$, if Y is the set of neighbors of X in G, then $|Y| \geq (1-\epsilon)|R|$.*
- **Explicitness.** *G can be constructed in $\mathrm{poly}(n, 1/\epsilon)$ time.*

The following theorem combines Lemma 3.1.3 and Lemma 3.1.4 to get the first component in our construction.

Theorem 3.1.5 *For every prime power q and $0 < \delta < 1$ there exist $0 < \rho < 1$, a positive integer β and a family of linear encoding functions $E_{C_k} : \mathbb{F}_q^k \to \mathbb{F}_q^{n(k)}$ satisfying the following requirements for all k:*

- **Rate.** *$k < n(k) \leq k/\rho$ and β divides $n(k)$.*
- **Relative β-distance.** *Parsing the output of C_k as $n(k)/\beta$ blocks of length β, the resulting code has relative distance $\geq \delta$.*

Furthermore, C_k can be computed by a uniform family of linear-size arithmetic circuits over F_q.

Proof: The construction combines a linear-time encoding function E_k of the $[\lfloor d_1 k \rfloor, k, \lceil d_2 k \rceil]_q$ code C'_k from Lemma 3.1.3 and the bipartite graph G with parameters $\epsilon = \min\{\frac{d_2}{d_1}, 1-\delta\}$ and $n = \lfloor d_1 k \rfloor$ from Lemma 3.1.4. To encode a message $x \in \mathbb{F}_q^k$ we compute $y = E_k(x) \in \mathbb{F}_q^n$ and redistribute the symbols according to G. More concretely, For every $i \in [n]$ and $j \in [\Delta]$ let $\gamma(i,j)$ be the index of the j'th neighbor of the i'th vertex in R. The $(i-1)\cdot\Delta + j$ entry of $E_{C_k}(x)$ is defined to be the $y_{\gamma(i,j)}$.

Since $\Delta = \lceil c_1/\epsilon^3 \rceil$ we have that $\frac{1}{d_1\Delta}$, the rate of C_k, is bounded away from zero. The linearity and linear encoding complexity of C_k are inherited from C'_k. For the relative block distance, let $\beta = \Delta$ and $x, x' \in \mathbb{F}_q^k$ such that $x \neq x'$. Since the relative distance of C'_k is at least $\frac{d_2}{d_1}$, there exists a set $X \subset A$ of size at least $\frac{d_2}{d_1}n$ such that for every $i \in X$ we have $E_k(x)_i \neq E_k(x')_i$. Since $\epsilon \leq \frac{d_2}{d_1}$ the expansion property of G guarantees that at least

$1 - \epsilon \geq \delta$ fraction of the blocks (of size $\beta = \Delta$) of $E_{C_k}(x_1)$ and $E_{C_k}(x_2)$ are different. ■

The linear-time computable extractor for bit-fixing sources we use generalizes the one from [25, Theorem 3.2] to arbitrary fields and is proved similarly. We rely on the following generalized transposition principle [10].

Lemma 3.1.6 (Implicit in [10],[25, Theorem 3.2]) *There exists a constant $c_4 > 0$ such that for every finite field $\mathbb{F}$ the following holds. Let C be an arithmetic circuit over $\mathbb{F}$ of size t which consists of only addition gates and computes the function $f(x) = Ax$. Then there exists an arithmetic circuit C' of size at most $c_4 t$ that computes the function $f'(x) = A^T x$.*

Applying Lemma 3.1.6 to the codes from Lemma 3.1.3 and using the known relation between extractors for bit-fixing sources and linear codes [14], we get the following.

Theorem 3.1.7 *There exist constants $c_2, c_3 > 0$ such that for every field $\mathbb{F}$ there exists a family of linear $(\lceil c_2 k \rceil, k, \lfloor c_3 k \rfloor)$ extractors for bit fixing sources over $\mathbb{F}$ that are computable by a uniform family of linear-size arithmetic circuits.*

We now show how to use the above components for the construction of linear uniform output families. The following construction takes an input length k and outputs a linear-size arithmetic circuit D_k representing a linear uniform output family from $\mathbb{F}^k$ to $\mathbb{F}^n$ where $n = \Theta(k)$.

Construction 3.1.8 *Parameter: prime power q. Input: positive integer k.*

Fix c_2 and c_3 as in Theorem 3.1.7 and δ such that $1 - c_2 < \delta < 1$. Let C be a code over $\mathbb{F}_q$ with message length k, block length $n(k)$, rate parameter ρ and relative β-distance $\geq \delta$ as promised by Theorem 3.1.5.

Let H be a random block diagonal matrix over $\mathbb{F}_q$ with $\frac{n(k)}{\beta}$ blocks each of size $\beta \times \beta$. H can be naturally identified with a vector $r \in \mathbb{F}_q^{\beta n(k)}$. We denote by H_r the matrix H corresponding to r.

Finally, let $k' = n(k)$ and let g be a linear $(\lceil c_2 k' \rceil, k', \lfloor c_3 k' \rfloor)$ extractor for bit fixing sources over $\mathbb{F}_q$ due to Theorem 3.1.7.

The output of the construction is the circuit $D_k : \mathbb{F}_q^k \times \mathbb{F}_q^{\beta n(k)} \to \mathbb{F}_q^{c_3 n(k)}$ computing $D_k(x, r) = g \circ H_r \circ E_C(x)$.

Lemma 3.1.9 *Construction 3.1.8 outputs a circuit* $D_k : \mathbb{F}_q^k \times \mathbb{F}_q^{m(k)} \to \mathbb{F}_q^{n'(k)}$ *such that* $m(k) = O(k)$,$n'(k) = \Omega(k)$, D_k *is of size* $O(k)$, *and* $A_r(x) = D_k(x, r)$ *defines a linear uniform output family when* r *is chosen uniformly from* $\mathbb{F}_q^{m(k)}$.

Proof: The length parameters, linearity, and linear circuit size of A_r are directly inherited by the building blocks. For the purpose of showing that the family is indeed uniform, let us view C as a code over $\Sigma = \mathbb{F}_q^\beta$ with relative distance $\geq \delta$. For any nonzero $x \in \mathbb{F}^k$ we consider $A_r x$, where r is chosen uniformly from $\mathbb{F}_q^{m(k)}$. Let L denote the places where $E_C(x)_i \neq 0$ and observe that the L-entries of $H_r(E_C(x))$ are distributed uniformly and independently whereas the rest of the entries are always zero. By the choice of C, the L-entries contain at least $\delta > 1 - c_2$ fraction of the symbols and by the choice of g having at least $1 - c_2$ fraction of the input being random is enough to ensure that the output is completely random. Together we have that $A_r x = D_k(x, r)$ is uniformly distributed. Finally, note that the size of the output is $c_3 n(k) > c_3 k$ as promised. ■

Note that padding the input with zeros and truncating the output allow us to respectively decrease and increase the ratio between k and n. The main theorem of this section thus follows.

Theorem 3.1.10 (Construction of Linear Uniform Output Families) *Let* $\mathbb{F}$ *be a finite field. For every positive integer* c *there is a polynomial-time constructible circuit family* $D_k : \mathbb{F}^k \times \mathbb{F}^{m(k)} \to \mathbb{F}^{ck}$ *such that* $m(k) = O(k)$, D_k *is of size* $O(k)$ *and* $A_r(x) = D_k(x, r)$ *defines a linear uniform output family when* r *is chosen uniformly from* $\mathbb{F}^{m(k)}$.

3.2 Applications in Coding Theory

Our interest in linear uniform output families is due to their pseudorandom nature. It is well known that a purely random (linear) function defines a good code with overwhelming probability. We show that the relaxed requirement of linear uniform output families is enough for them to match the rate/distance tradeoff achieved by purely random codes. We will also establish other useful properties to be used in Chapters 4 and 5.

We start by showing that a random member of a linear uniform output family meets the GV bound with high probability. This means that the best rate/distance tradeoff known for random linear codes is achieved by any linear uniform output family. Our proof is essentially the same as the one for random linear codes which can be found, e.g., in [19, Notes 2, Theorem 8], and is given here for self-containment. In fact, it is implicit in the original proof that the only properties needed from a random matrix A to define a code that meets the GV bound is its linearity and uniform output property (i.e. that Ax is uniformly distributed for a nonzero x).

Theorem 3.2.1 (GV Bound for Linear Uniform Output Families) *For every prime power q, $\delta \in [0, 1 - 1/q)$ and $\epsilon > 0$, the following holds for all positive integers n. Let $k = \lceil (1 - H_q(\delta) - \epsilon)n \rceil$. If $\mathcal{A}$ is a (q, k, n) linear uniform output family over $\mathbb{F}_q$, then the image of $A \in_R \mathcal{A}$ is a linear code with rate $R = k/n$ and relative distance at least δ, except with at most $q^{-\Omega(n)}$ probability (over the choice of A from $\mathcal{A}$).*

Proof: By the linearity of A both assertions are implied by showing that for any nonzero $x \in \mathbb{F}_q^k$ we have $\Delta(Ax) > \delta n$ except with exponentially small probability. Since $\mathcal{A}$ is a linear uniform output family, one has

$$\Pr[\Delta(Ax) \leq \delta n] \leq \frac{\mathrm{VOL}_q(n, \delta n)}{q^n}.$$

Recalling that

$$\mathrm{VOL}_q(n, \delta n) \leq q^{H_q(\delta)n},$$

a union bound over all $x \in \mathbb{F}_q^k$ yields

$$\Pr[\exists x \in \mathbb{F}_q^k \setminus \{0\} : \Delta(Ax) \leq \delta n)] \leq$$

$$q^k \frac{q^{H_q(\delta)n}}{q^n} \leq q^{(1-H_q(\delta)-\epsilon)n+1} q^{(H_q(\delta)-1)n} =$$

$$q \cdot q^{-\epsilon n}.$$

. ■

We will also need the following equivalent formulation.

Corollary 3.2.2 *Let $R \in (0,1)$ and let $\mathcal{A}$ be a (q, Rn, n) linear uniform output family. For every $\epsilon > 0$ a random $A \in_R \mathcal{A}$ defines a code with rate R and relative distance at least $H_q^{-1}(1-R) - \epsilon$ with probability $1 - q^{-\Omega(n)}$.*

We next show that the uniform output property is symmetric in the sense that a uniform output family has uniform output when multiplied from the left by any nonzero vector. This will allow us to deduce that the dual codes of codes obtained by uniform output families are good on their own right. This property of the dual codes will be useful for the application to secret sharing discussed in the next chapter.

Lemma 3.2.3 *If $\mathcal{A}$ is a linear uniform output family then for any nonzero $y \in \mathbb{F}_q^n$ and $z \in \mathbb{F}_q^k$ it holds that*

$$\Pr_{A \in_R \mathcal{A}}[y^T A = z] = q^{-k}.$$

Proof: Let x, y be nonzero vectors in $\mathbb{F}_q^k$ and $\mathbb{F}_q^n$ respectively. The uniform output of $\mathcal{A}$ ensures us that Ax is uniformly distributed whenever A is chosen according to $\mathcal{A}$. The fact that y is nonzero thus implies that $(y^T A)x = y^T(Ax)$ is uniform as well. Recalling that this holds for every nonzero x allows us to apply the XOR-lemma (Lemma 2.4.1) on $y^T A$ which completes the proof. ■

Lemma 3.2.4 *Let $R \in (0,1)$ and let $\mathcal{A}$ be a (q, Rn, n) linear uniform output family. For every $\epsilon > 0$ a random $A \in_R \mathcal{A}$ defines a code D whose dual code $D^\perp$ has rate $R^\perp = 1 - R$ and relative distance at least $H_q^{-1}(1-R^\perp) - \epsilon$ with probability $1 - q^{-\Omega(n)}$.*

Proof: By the above lemma and the fact that $D^\perp = \{y : y^T A = 0\}$ one has $\Pr[y \in D^\perp] \leq q^{-k}$ where $k = Rn$. Following the exact proof outline of Theorem 3.2.1 we get that for every $\epsilon > 0$, $D^\perp$ has relative distance $H_q^{-1}(1 - (n-k)/n) - \epsilon$ with probability $1 - q^{-\Omega(n)}$. Observing that the dimension $k^\perp$ of $D^\perp$ is $n - k$ with probability $1 - q^{-\Omega(n)}$ completes the proof. ■

The following theorem follows directly from the previous lemmata and will be used as the main tool in Chapter 4.

Theorem 3.2.5 (Simultaneous GV Bound) *Let $R \in (0,1)$ and $A \in_R \mathcal{A}$ be a random member of a (q, Rn, n) linear uniform output family. Let $D = Im(A)$ be the code associated with the generator matrix A and $k = Rn$. Then for every $\epsilon > 0$ the following holds with probability $1 - q^{-\Omega(n)}$:*

- *D has relative distance $\geq H_q^{-1}(1 - k/n) - \epsilon$ and dimension k.*
- *$D^{\perp}$ has relative distance $\geq H_q^{-1}(1 - (n-k)/n) - \epsilon$ and dimension $k^{\perp} = n - k$.*

Chapter 4

Applications in Information-Theoretic Cryptography

In this chapter we apply our new code family towards a linear-time implementation of secret sharing, a central tool in information-theoretic cryptography.

A secret sharing scheme [37, 6] allows a dealer to share a secret among n players so that only certain subsets of players can reconstruct the secret and others learn nothing about it. The typical case of interest is that of threshold secret sharing, where every set of at most t players learn nothing about the secret and every larger set of players can jointly reconstruct it. We consider the natural relaxation of *near-threshold* secret sharing, where there may be a small gap between the secrecy and reconstruction thresholds. More concretely, an (n, t, r) secret sharing scheme is an algorithm which given a secret s (a single bit by default) and a sequence of random bits produces an n-tuple of strings, called shares, such that any t shares reveal nothing about s and any r shares can be used to reconstruct s. The scheme is called an *ϵ-threshold* scheme if $t \geq (1-\epsilon)(r-1)$ and simply a *threshold* scheme when $t = r - 1$.

We start by reviewing some known connections between linear codes and secret sharing schemes. We give a self-contained general transformation from codes to secret sharing and tie the underlying code's parameters with

the parameters of the resulting secret sharing scheme. We then prove that by plugging our codes into that transformation one obtains an ϵ-threshold scheme for every arbitrarily small $\epsilon > 0$ such that the secret sharing phase and reconstruction can be performed by linear-size circuits.

4.1 From Codes to Secret Sharing

The relation between codes and secret sharing has been discovered by [33] and relies on the following observation. Given a code C, a codeword $c \in C$ and access to enough entries of c, one may reconstruct all the entries of c correctly. On the other hand, for some choices of C, access to a small amount of entries of c gives us no information about the other entries. Thus, to share a secret s one may choose a codeword c whose last entry is s and distribute the rest of the entries of c among the different players. This idea is the basis for the construction of Massey [31, 32] which was further explored by Chen et al. [12]. We now give a formal self-contained description of a generic code to secret sharing transformation which strongly relies on these results.

4.1.1 The Transformation

Let C be a $[n+1, k, d]_q$ code over $\mathbb{F}_q$ and consider the following secret sharing scheme for n players.

Algorithm 1 Share ($s \in \mathbb{F}_q$)

Choose a random $c \in C$ such that $c_{n+1} = s$
$\forall i \in [n] : s_i \leftarrow c_i$

Choosing such a c can be done in the following manner. Fix a global $c^* \in C$ with $c^*_{n+1} \neq 0$ and choose a random $c' \in C$. Set $c = c' + \alpha \cdot c^*$ where $\alpha \in \mathbb{F}_q$ is chosen such that $c_{n+1} = s$, i.e. $\alpha = \frac{s - c'_{n+1}}{c^*_{n+1}}$. To do this in linear-time when C is a code from our family with an encoding matrix $A \in \mathcal{A}$ one need only choose a random $x \in \mathbb{F}_q^k$ and compute $c' = Ax$ using the corresponding linear size circuit.

Algorithm 2 Reconstruct $R = (s_i)_{i \in I}$

Find $c' \in C$ such that $\forall i \in I : s_i = c'_i$
Return c'_{n+1}

Note that finding such a c can be done by solving the corresponding linear system.

Lemma 4.1.1 *If $|I| \geq n - d + 2$ then Reconstruct(R)=s.*

Proof: Since R is missing at most $d - 1$ pieces of c there is a unique c' which agrees with c on the indices of I, namely c itself. ■

The following lemma shows that a set of at most $d^{\perp} - 2$ players learn nothing about the secret.

Lemma 4.1.2 *If $|I| \leq d^{\perp} - 2$ then the distribution over the possible values of $R = (s_i)_{i \in I}$ induced by Share(s) is independent of s.*

Proof: We first show that any possible sequence R does not determine the value of s. Indeed, if the opposite is correct then one has the relation $x_{n+1} = \sum_{i \in I} \alpha_i x_i$ for some field elements $(\alpha_i)_{i \in I}$ and hence $\sum_{i \in I} \alpha_i x_i - x_{n+1} = 0$ is one of the constraints that defines C as a linear subspace. Recalling that $C^{\perp}$ is exactly the set of all such constraints and that $|I| \leq d^{\perp} - 2$ shows that the minimal distance of $C^{\perp}$ is at most $d^{\perp} - 1$, a contradiction.

Let $K = \{c_i = s_i | i \in I\}$ be the set of constraints associated with a feasible sequence R. The above shows that these constraints, put together with those that define C, do not determine the value of c_{n+1} and since R is feasible they must have a freedom degree $l > 0$. Thus, setting c_{n+1} to be an arbitrary value $s' \in \mathbb{F}_q$ leaves us with exactly q^{l-1} codewords which agree with both R and $c_{n+1} = s'$, as promised. ■

Corollary 4.1.3 *The general transformation applied on an $[n+1, k, d]_q$ code C yields a $(n, t, n - d + 2)$ secret sharing scheme where $t \geq d^{\perp} - 2$.*

4.2 Near Threshold Schemes with Linear Circuit Complexity

Theorem 4.2.1 (Secret sharing by linear-size circuits) *Let c_s (relative secrecy threshold) and c_r (relative reconstruction threshold) be constants such that $0 < c_s < c_r < 1$. Then for any n there is a circuit D_n of size $O(n)$, which on input $s \in \{0,1\}$ and a random $r \in_R \{0,1\}^{O(n)}$ outputs an n-tuple of shares $(s_1, \ldots, s_n)$ where $s_i \in \{0,1\}^{O(1)}$ such that:*

- **c_s-secrecy.** *The joint distribution of every set of at most $c_s n$ shares is independent of s (and in particular, reveals nothing about s).*
- **c_r-reconstruction.** *The secret s can be recovered from any set of at least $c_r n$ shares.*

Furthermore, there exists a PPT algorithm Gen *which on input 1^n outputs a circuit D_n as above along with a corresponding reconstruction circuit except with $2^{-\Omega(n)}$ failure probability.*

Proof: Let $\epsilon > 0$ be a small constant to be fixed later and let q be large enough power of 2 such that for every $x \in [0,1]$ it holds that $H_q^{-1}(x) \geq x - \epsilon/2$. In this case, Theorem 3.2.5 insures that except for a $2^{-\Omega(n)}$ failure probability, a code C defined by a random generator matrix $A \in_R \mathcal{A}$ where $\mathcal{A}$ is a $(q, k = Rn, n+1)$ linear uniform output family due to Theorem 3.1.10 has relative distance $\delta \geq H_q^{-1}(1-R) - \epsilon/2 \geq 1 - R - \epsilon$ and dual relative distance $\delta^\perp \geq H_q^{-1}(1-R^\perp) - \epsilon/2 = H_q^{-1}(R) \geq R - \epsilon$. By using the general transformation we get a $(n, \lceil k - \epsilon n \rceil, \lfloor k + \epsilon n + 2 \rfloor)$ scheme over $\mathbb{F}_q$. Choosing $k = \lfloor (c_s + c_r)n/2 \rfloor$ and a small enough value for ϵ we get

$$\lfloor k + \epsilon n + 2 \rfloor \leq ((c_s + c_r)/2 + \epsilon)n + 2 < c_r n$$

reconstruction and

$$\lceil k - \epsilon n \rceil \geq ((c_s + c_r)/2 - \epsilon)n - 1 > c_s n$$

secrecy. The other assertions follow from the construction of our codes. ■

Remark 4.2.2 (On the efficiency of reconstruction) The circuit D_n computes a linear function of the secret and randomness. Hence, for every

fixed reconstructing set I reconstruction can be performed by a linear size circuit.

Chapter 5

Applications in Complexity-Based Cryptography

This chapter demonstrates the potential usefulness of our codes for linear-time implementations of cryptographic primitives that rely on computational intractability. Due to the difficulty in proving unconditional lower bounds for any relevant computational model, the security of such primitives must be based on plausible (but unproven) intractability assumptions.

The intractability assumptions we employ are variants of the well studied intractability assumptions related to the learning parity with noise (LPN) problem. Roughly speaking, the LPN problem says that for a random generator matrix A, a random message x, and a low-weight random noise vector e, it is hard to decode x from $(A, Ax + e)$ except with negligible probability. We observe that LPN-based cryptographic constructions can be extended to other distributions over matrices A, provided that the corresponding intractability assumption still holds and the distribution satisfies a mild additional structural requirement. Combined with our new families of codes, this can yield improved computation time (linear instead of quadratic) with similar security. An advantage of our approach is its win-win flavor: refuting the underlying assumption would imply interesting progress in coding theory.

We introduce the relevant definitions in Section 5.1. A search to decision

reduction for our variant of LPN is presented in Section 5.2. This reduction is essential to Section 5.3 in which we state and discuss our intractability assumption in further detail. Sections 5.4 and 5.5 will show how to use our codes to construct a symmetric encryption and identification schemes whose security is based on our new intractability assumption.

5.1 Preliminaries

Definition 5.1.1 *For a distribution $\mathcal{A}$ over $n \times k$ binary matrices, a noise parameter μ and $x \in \{0,1\}^k$ we define the distribution $\mathcal{A}_{x,\mu}$ over $\mathbb{F}^{n\times k} \times \mathbb{F}^n$ where $(A, Ax+e)$ has the same probability as choosing A according to $\mathcal{A}$ and e according to BER^n_μ. The distribution $\mathcal{A} \times U_n$ is denoted by $\mathcal{A}_u$.*

Definition 5.1.2 (Intractability) *For a family of distribution $\mathcal{L} = (\mathcal{A}^n)_{n=1}^\infty$ over $n\times k(n)$ binary matrices and a noise parameter μ the problem $\text{CODE}(\mathcal{L}, \mu)$ is defined as follows.*

- *Input: 1^n, oracle access to $\mathcal{A}^n_{x,\mu}$ where $x \in_R \{0,1\}^{k(n)}$ is a fixed random vector.*
- *Output: x.*

We say that $\text{CODE}(\mathcal{L}, \mu)$ is $(s(n), t(n), \epsilon(n))$ search secure *if every nonuniform adversary D of size $O(t(n))$ that makes $O(s(n))$ oracle queries solves the problem with probability at most $\epsilon(n)$. It is* search intractable *if every nonuniform polynomial size adversary D solves the problem with probability negligible in n.*

We say that $\text{CODE}(\mathcal{L}, \mu)$ is $(s(n), t(n), \epsilon(n))$ decision secure *if for every nonuniform adversary D of size $O(t(n))$ that makes $O(s(n))$ oracle queries it holds that*

$$|\Pr[x \in_R \{0,1\}^k : D^{\mathcal{A}^n_{x,\mu}}(1^n) = 1] - \Pr[D^{\mathcal{A}^n_u} = 1]| \leq \epsilon(n).$$

We say that $\text{CODE}(\mathcal{L}, \mu)$ is decision intractable *if for every nonuniform polynomial size adversary D the following value is negligible in n*

$$|\Pr[x \in_R \{0,1\}^k : D^{\mathcal{A}^n_{x,\mu}}(1^n) = 1] - \Pr[D^{\mathcal{A}^n_u} = 1]|.$$

We will often omit the superscript n when it is clear from context.

5.2 Search to Decision

By our terminology, the LPN problem with parameters $R \in (0,1)$ and $\mu \in 0, 1/2)$ can be expressed as $\text{CODE}(\mathcal{L}, \mu)$ where $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty} = (U_{n \times Rn})_{n=1}^{\infty}$. Put in words, it is the problem of extracting a random Rn-bit vector x given access to samples of the form $(A, Ax + e)$ where A is a random $n \times k$ binary matrix and e is sampled from BER_{μ}^{n}. One of the useful features of this problem (see [7]) is that it admits a search to decision reduction: the ability to distinguish between $(A, Ax + e)$ samples and (A, U_n) samples is enough to extract x from $(A, Ax + e)$. This allows one to assume a more plausible hardness assumption – the hardness of decoding a random linear code – whose refutation would imply a breakthrough in coding theory and then rely on the pseudorandomness of $(A, Ax + e)$ for the construction of cryptographic primitives (see [4, 16, 24] for some examples).

Unfortunately, it seems that known search to decision techniques do not apply for our new family of codes. However, by applying a simple transformation on our codes we show that in some useful sense these codes can be subjected to a search to decision reduction after all. We now elaborate and establish a search to decision reduction for our codes. The usefulness of this reduction will become clear in Section 5.3 where we discuss the intractability of our codes.

More concretely, we show a general transformation from a distribution over $n \times k$ matrices $\mathcal{A}$ to a distribution over $n \times (k+1)$ matrices $\tilde{\mathcal{A}}$ such that the search problem over $\tilde{\mathcal{A}}$ can be reduced to the decision problem over $\mathcal{A}$.

For a $n \times k$ matrix $A = (a_1, \ldots, a_k)$, $i \in [k+1]$ and $r \in \{0,1\}^n$ we denote by $A_{i,r} = (a_1, \ldots a_{i-1}, r, a_i, \ldots, a_k)$ the matrix A with the vector r injected at the i'th position. For a distribution $\mathcal{A}$ over $n \times k$ matrices we define a new distribution $\tilde{\mathcal{A}}$ over matrices of the form $A_{i,r}$ where A is chosen according to $\mathcal{A}$ and i, r are chosen in a uniform manner.

We will need to following technical lemmata. The first is a simple worst-case to average-case reduction for the problem of distinguishing $\mathcal{A}_{x,\mu}$ from $\mathcal{A}_u$. The second is a search to (worst-case) decision reduction which is partially based on ideas from [35, Lemma 4.2] and [26, Lemma 1].

Lemma 5.2.1 (Decisional Worst-Case to Average-Case) *Let $\mathcal{A}$ be a $n \times k$ matrix distribution such that the function $f : \mathcal{A} \times \{0,1\}^k \to \{0,1\}^n$ defined by $f(A, x) = Ax$ can be computed by a circuit of size t'. Assume that*

there exists a circuit D of size t making s oracle queries such that

$$|\Pr[x \in_R \{0,1\}^k : D^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[D^{\mathcal{A}_u} = 1]| \geq \delta.$$

Then there exists a circuit $\hat{D}$ of size $\hat{t} = O(t + st')$ making s oracle queries such that for every $x \in \{0,1\}^k$ it holds that

$$|\Pr[\hat{D}^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[\hat{D}^{\mathcal{A}_u} = 1]| \geq \delta.$$

Proof: To see that Algorithm 3 is as promised we need to observe that it transforms $(A, Ax + e)$ samples into $(A, Ax + e + Ar) = (A, A(x + r) + e)$ samples, that is, $\mathcal{A}_{x,\mu}$ into $\mathcal{A}_{x+r,\mu}$. On the other hand, $\hat{D}$ transforms $\mathcal{A}_u$ into itself. Hence for any $x \in \{0,1\}^k$ we have

$$\Pr[\hat{D}^{\mathcal{A}_{x,\mu}}(1^n) = 1] = \Pr[r \in_R \{0,1\}^k : D^{\mathcal{A}_{x+r,\mu}}(1^n) = 1] =$$

$$\Pr[r \in_R \{0,1\}^k : D^{\mathcal{A}_{r,\mu}}(1^n) = 1]$$

and

$$\Pr[\hat{D}^{\mathcal{A}_u} = 1] = \Pr[D^{\mathcal{A}_u} = 1]$$

which completes the proof. ■

Algorithm 3 $\hat{D}(1^n)$

$r \leftarrow_R \{0,1\}^k$
Collect s oracle samples (A, y)
Simulate D. Answer each oracle query by transforming a sample (A, y) into $(A, y + Ar)$

Remark 5.2.2 *The size of $\hat{D}$ is $\hat{t} = O(t + sn)$ when $\mathcal{A}$ is a linear uniform output family obtained by our construction.*

Lemma 5.2.3 *Let $\mathcal{A}$ be a $n \times k$ matrix distribution and assume that there exists a circuit D of size t making s oracle queries such that for every $x \in \{0,1\}^k$ it holds that*

$$|\Pr[D^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[D^{\mathcal{A}_u} = 1]| \geq \delta.$$

Then for $m = \delta^{-2} sk \log k$ there exists a circuit $\tilde{D}$ making $\tilde{s} = O(km \log m)$ oracle queries of size $\tilde{t} = O(t\tilde{s})$ such that for every $x \in \{0,1\}^{k+1}$ it holds that

$$|\Pr[\tilde{D}^{\tilde{\mathcal{A}}_{x,\mu}}(1^n) = x] \geq 1/2.$$

Proof: We let x^i denote the vector x after the removal of its i'th coordinate. The main idea is to extract the i'th bit of x by transforming the distribution $\tilde{\mathcal{A}}$ into either $\mathcal{A}_u$ or $\mathcal{A}_{x^i,\mu}$ depending on the value of x_i. Distinguishing between the two cases is then done by simulating D.

We start off by showing that choosing $l = O(\delta^{-2} \log k)$ for Algorithm 4 implies that $\tilde{D}_i(1^n) = x_i$ except with failure probability at most $1/10k$. Let p_u and p_{x^i} denote the values $\Pr[D^{\mathcal{A}_u} = 1]$ and $\Pr[D^{\mathcal{A}_{x^i,\mu}}(1^n) = 1]$ respectively. When $x_i = 0$ we have $y = A_{i,r}x = Ax^i + x_i \cdot r = Ax^i$ and hence $(A, y + r)$ is distributed according to $\mathcal{A}_u$ meaning that p and p' are both estimations of $\Pr[D^{\mathcal{A}_u} = 1]$. By the Chernoff bound taking l as above gives $|p - p'| \geq \delta/2$ (a wrong answer) with probability $O(1/k)$. On the other hand, when $x_i = 1$ we have $y = A_{i,r}x = Ax^i + x_i \cdot r = Ax^i + r$ and hence $(A, y + r)$ is distributed according to $\mathcal{A}_{x^i,\mu}$ meaning that p and p' are estimations of $\Pr[D^{\mathcal{A}_{x^i,\mu}}(1^n) = 1]$ and $\Pr[D^{\mathcal{A}_u} = 1]$ respectively. Another invocation of the Chernoff bound together with the fact that $|\Pr[D^{\mathcal{A}_{x^i,\mu}}(1^n) = 1] - \Pr[D^{\mathcal{A}_u} = 1] \geq \delta$ shows that $|p - p'| < \delta/2$ with probability at most $1/10k$.

A union bound over all $i \in [k+1]$ shows that we can recover x correctly with constant probability. The additional overhead is due to the fact that we need to collect $sl = O(s\delta^{-2} \log k)$ samples for each i. Greedily waiting for the needed sample by collecting at most $\log m$ samples (and outputing a random answer if the needed sample did not appear) adds at most $1/10k$ error probability for each i. ■

The following search to decision theorem is a direct consequence of the above lemmas.

Theorem 5.2.4 *A matrix distribution $\mathcal{A}$ is decision intractable iff the distribution $\tilde{\mathcal{A}}$ is search intractable.*

Theorem 5.2.4 will allow us to make win-win flavored hardness assumptions on our codes towards their usage for complexity-based cryptographic applications. The exact statement we use is the next corollary.

Algorithm 4 $\tilde{D}_i(1^n)$

$real, rand \leftarrow 0$
for all $j \in [l]$ **do**
  Collect s samples of the form $(A_{i,r}, y)$
  Simulate D. Answer each oracle query by transforming a sample $(A_{i,r}, y)$ into $(A, y + r)$
  If D outputs 1 then $real \leftarrow real + 1$
  Simulate D answering oracle queries according to $\mathcal{A}_u$
  If D outputs 1 then $rand \leftarrow rand + 1$
end for
Let $p = \frac{real}{l}, p' = \frac{rand}{l}$ be the estimations of the accept probability of D in the above simulations
Output 1 if $|p - p'| \geq \delta/2$, otherwise output 0

Corollary 5.2.5 *Let $\mathcal{A}$ be a linear uniform output family for which the function $f(A, x) = Ax$ has linear circuit complexity. Then $\mathcal{A}$ is decision intractable iff $\tilde{\mathcal{A}}$ is search intractable. Moreover, $\tilde{\mathcal{A}}$ is a linear uniform output family and has linear circuit complexity as well.*

5.3 Intractability

In Sections 5.4 and 5.5 we give some examples of how LPN based cryptographic construction can be generalized to any (not necessarily uniform) matrix distribution and consider their instantiation with our new codes as the underlying distribution. The security of these construction will follow directly from the decision intractability of the underlying codes. The advantage of using our new codes instead of completely random is in that the encoding and representation complexity of our codes is linear. This often translates to a linear-time and communication complexities for the primitive we are trying to achieve. Of course, it is often the case that a linear-time complexity can be achieved by paying a price in the primitive security and hence, one should simultaneously consider the construction's complexity and security when comparing different constructions.

To obtain meaningful results, we would like to argue that the decision problem $\mathrm{CODE}(\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}, \mu)$ with $\mathcal{A}^n$ being a $n \times Rn$ linear uniform output family according to our construction is as hard as the decision problem

$\mathrm{CODE}(\mathcal{R} = (\mathcal{R}^n)_{n=1}^{\infty}, \mu)$ where $\mathcal{R}^n$ is the uniform distribution over $n \times Rn$ matrices. One reason to believe such an assumption has to do with the random nature of our codes. The uniform output of our codes gurantees that $Ax + e$ defines the exact same distribution regardless of whether $A \in_R \mathcal{A}^n$ is chosen according to our distribution or is chosen to be a completely random matrix. It also seems that the randomness used by our construction together with the other components yields a family of matrices with no clear structure. Indeed, our best attempts to search or decide these families has consistently proved futile. It also seems to be the case that similar hardness assumptions for more structured distributions such as the family of Toeplitz matrices [17] (which also have the uniform output property but only known to be encodable in quasi-linear-time) are yet to be refuted.

Finally, our search to decision reduction gives a win-win flavor to the above assumption. As shown in Sections 5.4 and 5.5, the decision intractability of $\mathcal{L}$ implies the security of our generalized LPN based contructions where $\mathcal{L}$ is used as the underlying distribution. Furthermore, if we assume that searching $\mathcal{L}$ is as hard as searching the uniform distribution then our construction have a reduced linear-time complexity while enjoying the same security gurantees. On the other hand, if one can decide $\mathcal{L}$ then by our search to decision reduction, one can also search $\tilde{\mathcal{L}} = (\tilde{\mathcal{A}}^n)_{n=1}^{\infty}$. Since $\tilde{\mathcal{L}}$ is also encodable by linear-size circuits and has uniform output, this implies the exsitence of linear-time encodable and efficiently decodable codes that meet the GV bound, an unknown coding theoretic result.

For concreteness sake, we will measure our results against the state of the art LPN attacks. The best known algorithm due to Blum et al. [8] has time and query complexity $2^{O(n/\log n)}$. When insisting on a polynomial query complexity the best known running time is $2^{O(n/\log\log n)}$ [27, 29]. For constant query complexity, the problem is only known to be solved in exponential time.

5.4 Symmetric Encryption

Definition 5.4.1 (Symmetric Encryption) *Let $l = l(n)$ be a message length parameter. We say that $\Pi = (\textsc{Gen}, \textsc{Enc}, \textsc{Dec})$ is an $(s(n), t(n), \epsilon(n))$ secure stateless symmetric encryption scheme if the following holds:*

- *Correctness. For every n and $m \in \{0,1\}^{l(n)}$ we have*

$$\Pr[k \leftarrow \textsc{Gen}(1^n); c \leftarrow \textsc{Enc}(k,m); m' \leftarrow \textsc{Dec}(k,c) : m = m'] = 1$$

- *Security. For every nonuniform polynomial time adversary D of size at most $t(n)$,*

$$\Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] \leq 1/2 + \epsilon(n)$$

where the experiment $\mathsf{E}_{\mathsf{D}}(1^n)$ is defined as follows:

  - *$D(1^n)$ outputs a pair of messages vectors $\vec{M}_0 = (m_0^1, \ldots, m_0^{s(n)})$ and $\vec{M}_1 = (m_1^1, \ldots, m_1^{s(n)})$ where $m_b^i \in \{0,1\}^{l(n)}$*
  - *$k \leftarrow_R \textsc{Gen}(1^n)$; $b \leftarrow_R \{0,1\}$*
  - *For $i \in [s(n)]$, $c^i \leftarrow_R \textsc{Enc}(k, m_b^i)$*
  - *$b' \leftarrow_R D(c^1, \ldots, c^{s(n)})$*
  - *The output of the experiment E_{D} is defined to be 1 if $b' = b$, and 0 otherwise*

We present a symmetric encryption scheme construction based on the hardness of our codes. The construction enjoys linear (in the message length) size secret key as well as linear size encryption and decryption circuits and hence achieves the best efficiecy/security tradeoff currently known for LPN-based symmetric encryption schemes. Our construction is a straightforward generalization of [4, Construction 7] and [16, Figure 1] for general matrix distributions.

Construction 5.4.2 *Let $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$ be a distribution over $n \times k(n)$ matrices, $l = l(n)$ a message length parameter, $\mu \in (0, 1/2)$ a noise parameter and $\mathcal{G} = (G_n)_{n=1}^{\infty}$ a family of rate $R_n = l(n)/n$ error correcting codes generator matrices. The private key x is a random k bit vector.*

- *Encryption. To encrypt a message $m \in \{0,1\}^{l(n)}$, choose a matrix $A \in_R \mathcal{A}^n$ and a random noise vector $e \in \textsc{Ber}_{\mu}^n$. Output to ciphertext*

$$(A, Ax + e + G_n m).$$

- *Decoding. Given a ciphertext (A, y), apply the decoding algorithm of the code G_n on the vector $y - Ax$.*

We denote this scheme by $\mathrm{SE}(\mathcal{L}, \mu, \mathcal{G})$.

The next lemma ties the security of $\mathrm{SE}(\mathcal{L}, \mu, \mathcal{G})$ with the decision intractability of $\mathcal{L}$. Note that the lemma does not take the scheme's correctness into consideration.

Lemma 5.4.3 *Let $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$ be a family of $n \times Rn$ distributions, $\mu \in (0, 1/2)$ be a noise parameter and $\Pi = \mathrm{SE}(\mathcal{L}, \mu, \mathcal{G})$ such that $\mathcal{G}$ is encodable in time $t'(n)$. Let D be an adversary of size $t(n)$ making $s(n)$ oracle queries such that for infinitely many n*

$$\Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] > 1/2 + \epsilon(n).$$

Then there exists an adversary D' of size $O(t(n) + s(n) \cdot t'(n))$ making $s(n)$ oracle queries such that for infinitely many n

$$\Pr[x \in_R \{0,1\}^k : D'^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[D'^{\mathcal{A}_u} = 1] > \epsilon(n).$$

Proof: We define an adversary D' having oracle access to samples of the form (A, y) to simulate D with $\vec{c} = ((A_1, y_1 + G_n m_b^1), \ldots, (A_t, y_t + G_n m_b^{s(n)}))$ where b is a random bit and all the (A_i, y_i) are fresh oracle samples.

It is a straightforward matter to validate that

$$\Pr[x \in_R \{0,1\}^k : D'^{\mathcal{A}_{x,\mu}}(1^n) = 1] = \Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] > 1/2 + \epsilon(n),$$

whereas

$$\Pr[D'^{\mathcal{A}_u} = 1] = 1/2.$$

Putting the two together we get that for infinitely many n

$$\Pr[x \in_R \{0,1\}^k : D'^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[D'^{\mathcal{A}_u} = 1] > \epsilon(n).$$

■

By plugging in our new family of codes we obtain a new symmetric encryption scheme candidate enjoying the best known efficiency/security tradeoff.

Theorem 5.4.4 (Symmetric Encryption) *For $R, \mu \in (0, 1)$ and $l(n)$ let $\mathcal{A}^n$ be our linear uniform output family over $n \times Rn$ matrices and $\mathcal{L} =$*

$(\mathcal{A}^n)_{n=1}^{\infty}$. *Let* $\mathcal{C}$ *be a code family with rate* $l(n)/n$ *capable of fixing* $\rho > \mu$ *fraction of errors which is encodable and decodable by linear-size circuits. Then there exists a symmetric encryption scheme* $\Pi = (\textsc{Gen}, \textsc{Enc}, \textsc{Dec})$ *with the following properties:*

- $\textsc{Gen}, \textsc{Enc}$ *and* $\textsc{Dec}$ *can be implemented by linear-size circuits which are poly-time constructible.*
- *If* $\mathrm{CODE}(\mathcal{L}, \mu)$ *is* $(s(n), t(n) + n \cdot s(n), \epsilon(n))$ *decision secure then* Π *is* $(s(n), t(n), \epsilon(n))$ *secure.*

Proof: We only need to put all the pieces together. Let $\mathcal{G}$ be a family of generator matrix for the code $\mathcal{C}$ and let $\Pi = \mathrm{SE}(\mathcal{L}, \mu, \mathcal{G})$ be the symmetric encryption scheme obtained by applying construction 5.4.2 to $\mathcal{L}$ and $\mathcal{G}$. The security of the scheme follows directly from Lemma 5.4.3 and the assertions on $\textsc{Gen}, \textsc{Enc}, \textsc{Dec}$ follow from the properties of our linear uniform output family and the assumption on $\mathcal{C}$. Finally, decryption is always correct whenever the relative weight of the noise e is less than ρ, this does not happen only with exponentially negligible in n probability. For perfect correctness one may choose $e \in \textsc{Ber}_{\mu}^{n}$ conditioned on $\Delta(e) < \rho/n$. Such a noise distribution is exponentially close to the original one and hence neligibly affects the scheme's security. ■

Considering the state of the art LPN attacks leads to the following corollary.

Corollary 5.4.5 *For* $R, \mu \in (0,1)$ *let* $\mathcal{A}^n$ *be our linear uniform output family over* $n \times Rn$ *matrices and* $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$. *Assuming that* $\mathrm{CODE}(\mathcal{L}, \mu)$ *is* $(2^{o(n/\log n)}, 2^{o(n/\log n)}, 2^{-\Omega(n)})$ *[*$(\mathrm{poly}(n), 2^{o(n/\log\log n)}, 2^{-\Omega(n)})$, $(1, 2^{o(n)}, 2^{-\Omega(n)})$*] decision secure then there exists an* $(2^{o(n/\log n)}, 2^{o(n/\log n)}, 2^{-\Omega(n)})$ *[*$(\mathrm{poly}(n), 2^{o(n/\log\log n)}, 2^{-\Omega(n)})$, $(1, 2^{o(n)}, 2^{-\Omega(n)})$*] symmetric encryption scheme* $\Pi = (\textsc{Gen}, \textsc{Enc}, \textsc{Dec})$. *Furthermore,* $\textsc{Gen}, \textsc{Enc}$ *and* $\textsc{Dec}$ *can be implemented by linear-size circuits which are poly-time constructible.*

5.5 Identification

Definition 5.5.1 (Identification) *An* $(s(n), t(n), \epsilon(n))$ *secure identification protocol is a tuple* $\Pi = (\textsc{Gen}, \textsc{Cha}, \textsc{Res}, \textsc{Ver})$ *where* $\textsc{Gen}, \textsc{Cha}, \textsc{Res}$

are PPT algorithms and VER *is a polynomial-time algorithm for which the following correctness and security conditions hold:*

- *Correctness. For any secret key and challenge, a response generated with the same secret key is always accepted. Formally:*

$$\Pr[k \leftarrow \text{GEN}(1^n); c \leftarrow \text{CHA}(k); a \leftarrow \text{RES}(c,k) : \text{VER}(a,c,k)] = 1.$$

- *Security against passive adversaries. Let O_k denote the distribution over challenge/response pairs (c,a) where $c \leftarrow \text{CHA}(k)$ and $a \leftarrow \text{RES}(c,k)$. For every nonuniform adversary D of size $O(t(n))$ making $O(s(n))$ oracle queries it holds that*

$$\Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] \leq \epsilon(n)$$

  where the experiment $\mathsf{E}_{\mathsf{D}}(1^n)$ is defined as follows:

  - $k \leftarrow \text{GEN}(1^n)$
  - $c^* \leftarrow \text{CHA}(k)$
  - $a^* \leftarrow D^{O_k}(c^*)$
  - *The output of the experiment is defined to be* 1 *if* $\text{VER}(a^*, c^*, k) = 1$

Our Identification scheme is in essence a generalization of the well known HB protocol [24, Protocol 1] for general code families. With resemblance to our symmetric encryption construction, the scheme enjoys linear (in the security parameter) circuit and communication complexities and hence achieves the best efficiency/security tradeoff currently known for LPN-based constructions.

Construction 5.5.2 *Let $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$ be a distribution over $n \times k(n)$ matrices and $\mu \in (0, 1/2)$ a noise parameter. We define the scheme* $\text{HB}(\mathcal{L}, \mu)$ *as follows.*

- $\text{GEN}(1^n)$ *outputs a random* $x \in_R \{0,1\}^{k(n)}$
- $\text{CHA}(x)$ *outputs a random* $A \in_R \mathcal{A}^n$
- $\text{RES}(x, A)$ *outputs* $y = Ax + e$ *where* $e \in \text{BER}_{\mu}^n$

- $\textsc{Ver}(y, x, A)$ *accepts if* $\Delta(y - Ax) \leq (1 + \epsilon')\mu n$ *where* ϵ' *is a small constant (which exact value is discussed shortly)*

We denote this scheme by $\mathrm{HB}(\mathcal{L}, \mu)$. The next lemma ties the security of $\mathrm{HB}(\mathcal{L}, \mu)$ with the decision intractability of $\mathcal{L}$.

Lemma 5.5.3 *Let* $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$ *be a sequence of linear uniform output families over* $n \times Rn$ *matrices and* $\mu < H(1-R)/2$ *be a noise parameter such that* $\mathrm{CODE}(\mathcal{L}, \mu)$ *is* $(s(n), t(n), \epsilon(n))$ *decision secure. Then there exists an* $(s(n), t(n), \epsilon(n))$ *secure identification scheme* $\Pi = (\textsc{Gen}, \textsc{Cha}, \textsc{Res}, \textsc{Ver})$.

Proof: Assume towards a contradiction that an adversary D of size $O(t(n))$ making $O(s(n))$ oracle queries exists such that for infinitely many n we have

$$\Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] > \epsilon(n).$$

We define an adversary D' having oracle access to samples of the form (A, y) to simulate D by answering oracle queries with its own oracle samples (A, y).

It is a straightforward matter to validate that

$$\Pr[x \in_R \{0,1\}^k : D'^{\mathcal{A}_{x,\mu}}(1^n) = 1] = \Pr[\mathsf{E}_{\mathsf{D}}(1^n) = 1] > \epsilon(n).$$

On the other hand, with access to $\mathcal{A}_u$ we provide D with samples (A, r) where r is random and independent of x and hence D has no information about x. Furthermore, from Corollary 3.2.2 the challenge A^* defines a code with relative distance greater then $H(1-R) > 2(1+\epsilon')\mu$ except for exponentially small in n failure probability. This means that from D's perspective, the correct response $A^*x + e$ can belong to any of the 2^{Rn} disjoint Hamming balls around codewords A^*x' and thus

$$\Pr[D'^{\mathcal{A}_u} = 1] = 2^{-\Omega(n)}.$$

Putting the two together we get that

$$\Pr[x \in_R \{0,1\}^k : D'^{\mathcal{A}_{x,\mu}}(1^n) = 1] - \Pr[D'^{\mathcal{A}_u} = 1] > \epsilon(n) - 2^{-\Omega(n)}$$

for infinitely many n in contradiction to the assumption that $\mathcal{L}$ is is $(s(n), t(n), \epsilon(n))$ decision secure. Note that the only constraints on ϵ' are that $\epsilon' > 0$ and $H(1 - R) > 2(1 + \epsilon')\mu$. ■

We now present the main theorem of this section

Theorem 5.5.4 (Identification) *For $R \in (0,1)$ and $\mu < H(1-R)/2$ let $\mathcal{A}^n$ be our linear uniform output family over $n \times Rn$ matrices and $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$. Then there exists an identification scheme $\Pi = (\text{GEN}, \text{CHA}, \text{RES}, \text{VER})$ with the following properties:*

- GEN, CHA, RES *and* VER *can be implemented by linear-size circuits which are poly-time constructible.*
- *if* $\text{CODE}(\mathcal{L}, \mu)$ *is* $(s(n), t(n), \epsilon(n))$ *decision secure then* Π *is* $(s(n), t(n), \epsilon(n))$ *secure.*

Proof: Let $\Pi = \text{HB}(\mathcal{L}, \mu)$ be the identification scheme obtained by applying construction 5.5.2 to $\mathcal{L}$. The security of the scheme follows directly from lemma 5.5.3 and the assertions on GEN, CHA, RES, VER follow from the properties of our linear uniform output family. Finally, an honest identifier responds correctly whenever the weight of the noise e is less than $(1+\epsilon')\mu n$, this does not happen only with exponentially negligible in n probability. For perfect correctness one may choose $e \in \text{BER}_{\mu}^{n}$ conditioned on $\Delta(e) < (1+\epsilon')\mu n$. Such a noise distribution is exponentially close to the original one and hence the scheme's security is only negligibly affected. ■

Corollary 5.5.5 *For $R \in (0,1)$ and $\mu \in (0,1)$ such that $\mu < H(1-R)/2$ let $\mathcal{A}^n$ be our linear uniform output family over $n \times Rn$ matrices and $\mathcal{L} = (\mathcal{A}^n)_{n=1}^{\infty}$. Assuming that $\text{CODE}(\mathcal{L}, \mu)$ is $(2^{o(n/\log n)}, 2^{o(n/\log n)}, 2^{-\Omega(n)})$ $[(\text{poly}(n), 2^{o(n/\log\log n)}, 2^{-\Omega(n)}), (1, 2^{o(n)}, 2^{-\Omega(n)}))]$ decision secure then there exists an $(2^{o(n/\log n)}, 2^{o(n/\log n)}, 2^{-\Omega(n)})$ $[(\text{poly}(n), 2^{o(n/\log\log n)}, 2^{-\Omega(n)}), (1, 2^{o(n)}, 2^{-\Omega(n)}))]$ identifiction scheme $\Pi = (\text{GEN}, \text{CHA}, \text{RES}, \text{VER})$. Furthermore,* GEN, CHA, RES *and* VER *can be implemented by linear-size circuits which are poly-time constructible.*

Chapter 6

Conclusion and Open Questions

In this work we put forward a new family of linear codes which shares several useful properties of random linear codes while supporting linear-time encoding. This family yields the first randomized constructions of linear-time encodable codes which meet the Gilbert-Varshamov bound. It also gives rise to several cryptographic applications, including the first nontrivial linear-time secret sharing schemes and new candidate constructions of symmetric encryption and identification schemes which may be conjectured to exhibit better efficiency/security tradeoffs than existing candidates.

We leave open several natural questions. A first question is to try and gain better understanding of the decoding complexity of codes from our family. Do such codes admit significantly better decoding algorithms than those known for random linear codes? Is there a different (randomized) construction of linear-time encodable binary codes which meet the GV bound and yet support efficient decoding, even only in the presence of a small constant fraction of random errors?

A second question is to get an explicit family of codes which match the parameters of our randomized construction. This question appears to be very difficult even without the linear-time encoding requirement. Indeed, the goal of obtaining an explicit family of binary codes which meet the GV bound is a longstanding problem, and in fact even the relaxed goal of meeting the GV bound using only $o(n)$ random bits seems challenging.

The applications of our new codes to complexity-based cryptography mainly rely on the conjectured intractability of decoding and are less sensitive to the exact rate/distance tradeoff. Are there simpler randomized constructions of linear-time encodable codes which may suffice for these applications?

Finally, it may be interesting to extend the application domain of our technique to include linear-time implementations of other pseudorandom objects. In particular, there are several applications of dimension reduction via linear functions (such as the Johnson-Lindenstrauss transform and related problems, cf. [2]) for which no linear-time implementation is known.

Bibliography

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *STOC*, pages 557–563, 2006.

[3] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992.

[4] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.

[5] Joshua Baron, Yuval Ishai, and Rafail Ostrovsky. On linear-size pseudorandom generators and hardcore functions. In *COCOON*, pages 169–181, 2013.

[6] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, Monval, NJ, USA, 1979. AFIPS Press.

[7] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1993.

[8] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[9] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pages 757–804. 1990.

[10] J. L. Bordewijk. Inter-reciprocity applied to electrical networks. *Applied Scientific Research B: Electrophysics, Acoustics, Optics, Mathematical Methods*, 6:1–74, 1956.

[11] Michael R. Capalbo, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *IEEE Conference on Computational Complexity*, page 15, 2002.

[12] Hao Chen, Ronald Cramer, Shafi Goldwasser, Robbert de Haan, and Vinod Vaikuntanathan. Secure computation from random error correcting codes. In *EUROCRYPT*, pages 291–310, 2007.

[13] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *FOCS*, pages 396–407, 1985.

[14] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem of t-resilient functions (preliminary version). In *FOCS*, pages 396–407, 1985.

[15] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952.

[16] H. Gilbert, M. J. Robshaw, and Y. Seurin. How to encrypt with the LPN problem. In *Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part II*, ICALP '08, pages 679–690, Berlin, Heidelberg, 2008. Springer-Verlag.

[17] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. HB$^{\#}$: Increasing the security and efficiency of HB^{+}. In *EUROCRYPT*, pages 361–378, 2008.

[18] Oded Goldreich, Hugo Krawczyk, and Michael Luby. On the existence of pseudorandom generators. In *CRYPTO*, pages 146–162, 1988.

[19] V. Guruswami. Introduction to coding theory - lecture notes, 2010.

[20] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*, volume 3282 of *Lecture Notes in Computer Science*. Springer, 2004.

[21] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *FOCS*, pages 658–667, 2001.

[22] Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting Gilbert-Varshamov bound for low rates. In *SODA*, pages 756–757, 2004.

[23] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[24] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *ASIACRYPT*, pages 52–66, 2001.

[25] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *STOC*, pages 433–442, 2008.

[26] J. Katz and J. S. Shin. Parallel and concurrent security of the HB and HB+ protocols. In *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pages 73–87, Berlin, Heidelberg, 2006. Springer-Verlag.

[27] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. *SIAM J. Comput.*, 42(3):1302–1326, 2013.

[28] Hugo Krawczyk. LFSR-based hashing and authentication. In *CRYPTO*, pages 129–139, 1994.

[29] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM*, pages 378–389, 2005.

[30] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The computational complexity of universal hashing. *Theor. Comput. Sci.*, 107(1):121–133, 1993.

[31] J. L. Massey. Minimal codewords and secret sharing. In *Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory*, pages 276–279, 1993.

[32] J. L. Massey. Some applications of coding theory in cryptography. In *Codes and Ciphers: Cryptography and Coding IV*, pages 33–47, 1995.

[33] Robert J. McEliece and Dilip V. Sarwate. On sharing secrets and reed-solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[34] S. I. Gelfand R. L. Dobrushin and M. S. Pinsker. On the complexity of coding. *Proc. 2nd Internat. Symp. on Information Theory*, pages 174–184, 1973.

[35] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[36] J. E. Savage. *The Complexity of Computing*. Krieger Publishing Co., Inc., Melbourne, FL, USA, 1987.

[37] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[38] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[39] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[40] M. A. Tsfasman. Goppa codes that are better than the Varshamov–Gilbert bound. *Probl. Peredachi Inf.*, 18:3–6, 1982.

[41] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akadamii Nauk*, 117:739–741, 1957.

קודים בעלי זמן קידוד לינארי וקריפטוגרפיה

ארז דרוק

קודים בעלי זמן קידוד לינארי וקריפטוגרפיה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

ארז דרוק

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
תשרי ה'תשע"ד חיפה ספטמבר 2013

המחקר נעשה בהנחיית פרופ' יובל ישי בפקולטה למדעי המחשב.

ברצוני להודות למנחה שלי, פרופסור יובל ישי, על שלימד אותי כיצד לערוך מחקר, על כך שסיפק את הרעיונות, התובנות, השאלות והתשובות העיקריים בעבודה זו ועל כך שתמיד מצא זמן וסבלנות אלי. ברצוני להודות לאריאל גביזון, אלעד הרמתי, אייל קושילביץ, נתנאל רביב, רוני רות' ואמיר יהודיוף על תרומתם. תודה גם לירדה קולט על יכולתה להפוך כל דבר לאפשרי. לבסוף, תודה למשפחתי היקרה על תמיכה ועזרה שלא ניתנות לערעור.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר

איכותו של קוד לתיקון שגיאות נקבעת בראש ובראשונה על פי שני מדדים בעלי אופי סותר. הראשון, מרחק הקוד, מעיד על יכולות תיקון וזיהוי השגיאות של הקוד בעוד השני, קצב הקוד, מעיד על כמות האינפורמציה הנישאת על ידי מילות הקוד השונות. למרות שניתן באופן טריוויאלי לשפר כל אחד מהמדדים הללו כרצוננו תוך כדי התעלמות מהשני, תוצאות רבות מראות כי שיפור של שני המדדים (או שיפור של אחד ושימור של האחר) היא משימה מורכבת ולפעמים בלתי אפשרית. יתר על כן, הבנה מלאה של יחסי הגומלין בין שני המדדים הללו עדיין נמצאת מחוץ להישג ידו של המדע. אחד מהיעדים הראשונים בדיסציפלינת הקודים לתיקון שגיאות היא בנייה של קודים טובים אסימפטוטית. קוד טוב אסימפטוטית הינו משפחה של קודים בה קצב הקוד וכן מרחקו היחסי חסומים הרחק מאפס.

עבור אובייקטים קומבינטורים רבים מתקיימת התכונה המעניינת הבאה. בניית האובייקט באמצעות תהליך אקראי מובילה, בהסתברות טובה, למימוש מוצלח של האובייקט. תכונה זו מתקיימת גם עבור קודים לתיקון שגיאות. משפחה אקראית של קודים, בין אם ליניאריים ובין אם לאו, מהווה קוד טוב אסימפטוטית בהסתברות גבוהה. למעשה, קודים כאלה משיגים את יחס הגומלין הטוב ביותר הידוע בין מרחק הקוד וקצב הקוד מעל השדה הבינארי. מעל שדות גדולים יותר, קודים אקראיים נחותים רק במקצת ביחס לקודים יותר מתוחכמים כגון [40]. חסם גילברט־ורשמוב [15, 41] מתאר באופן מדויק את יחס הגומלין מרחק/קצב של קודים אקראיים וקודים שמשיגים חסם זה נחשבים טובים מאוד באספקט הזה. בפרט, חסם גילברט־ורשמוב מראה כי קודים אקראיים הם קודים טובים אסימפטוטית בהסתברות גבוהה.

סיבוכיות הקידוד, גם היא פרמטר הקובע את איכות הקוד, מקבלת תשומת לב מיוחדת בעבודה זו. עיקר עניינינו יהיה בקודים בעלי זמן קידוד לינארי. על פי מדד זה, קודים אקראיים, למרות שמשיגים את חסם גילברט־ורשמוב נכשלים באופן חרוץ שכן זמן הקידוד שלהם מעריכי במקרה הכללי וריבועי במקרה הלינארי. סיבוכיות הקידוד האסימפטוטית הטובה ביותר שניתן לקוות לקבל היא לינארית ומכאן עולה באופן טבעי

השאלה האם קודים טובים אסימפטוטית בעלי זמן קידוד לינארי קיימים. תשובה חיובית לשאלה זו הוצגה לראשונה על ידי [34] אשר הציגו בנייה אקראית של קודים בינאריים בעלי זמן קידוד לינארי. בנייה מפורשת שבנוסף ניתנת לפיענוח בזמן לינארי הוצגה לראשונה בעבודתו הידועה של ספילמן [39].

הקודים של ספילמן, למרות היותם טובים אסימפטוטית, נמצאים רחוק מחסם גילברט־ורשמוב. גורוסואמי ויינדיק [23] מציגים קודים בעלי זמן קידוד לינארי המתקרבים לחסם גילברט־ורשמוב קרוב ככל שנחפוץ. למרבה הצער, על קירבה יותר טובה לחסם נאלץ לשלם מחיר בגודל השדה. תוצאות אלו משאירות את שאלת קיומם של קודים בעלי זמן קידוד לינארי המשיגים את חסם גילברט־ורשמוב או אפילו מתקרבים אליו מעל השדה הבינארי פתוחה. שאלה זו היא מקור המוטיבציה העיקרי לעבודה זו.

התרומה שלנו

אנו מציגים בנייה אקראית של קודים בעלי זמן קידוד לינארי הנהנים מכמה תכונות של קודים אקראיים לחלוטין. בפרט, הבנייה שלנו היא הראשונה המציגה קודים בעלי זמן קידוד לינארי המשיגים את חסם גילברט־ורשמוב ותוך כך עונה בחיוב על השאלה לעיל מעל כל שדה סופי. הבניה שלנו מסתמכת על אובייקט פסאודו־אקראי שנבחר להיקרא משפחה בעלת פלט אחיד. משפחה בעלת פלט אחיד היא משפחה של פונקציות לינאריות בעלת התכונה שעבור כל קלט שונה מאפס הפלט מפולג באופן אחיד בהינתן בחירה אקראית של פונקציה מהשפחה. פרק 3 מציג בנייה של אובייקט זה עבורה כל פונקציה במשפחה ניתנת לתיאור באורך לינארי וכן לחישוב בזמן לינארי. לאחר הצגת הבנייה אנו מציגים כמה קשרים בין משפחות אלו לקודים לתיקון שגיאות ובפרט בנייה של קודים בעלי זמן קידוד לינארי המשיגים את חסם גילברט־ורשמוב בעזרת המשפחה שלנו.

חיסרון בולט של הקודים שלנו נוגע לשאלת פיענוח הקוד. בניגוד לקודים של [23, 39] אנו לא יודעים האם ואיך ניתן לפענח את הקודים שלנו באופן יעיל. זוהי לא בהכרח בעיה משתי סיבות. הראשונה, קיימות אפליקציות מסויימות של קודים הדורשות יכולת פיענוח ממחיקות בלבד, זאת תמיד ניתן לבצע עבור קודים לינאריים ובפרט עבור הקודים שלנו. שנית, אם בעיית פיענוח הקוד הינה קשה חישובית (כפי שאנו מניחים), יש תקווה שקודים אלו יהיו שימושיים בתחום הקריפטוגרפיה. קיימות בניות קריפטוגרפיות לא מעטות המסתמכות על הנחת קושי הפיענוח של קוד לינארי אקראי. משפחת הקודים שלנו הינה מועמד למשפחה יותר מצומצת של קודים בעלת זמן קידוד וגודל תיאור לינאריים אשר במקביל משמרת את המרחק המינימאלי וקושי הפיענוח של

קודים לינאריים אקראיים לחלוטין.

פרקים 4 ו־ 5 עוסקים בשימושים של הקודים שלנו בתחום הקריפטוגרפיה. אנו עושים הבחנה בין קריפטוגרפיה מבוססת אינפורמציה בה ההגנה הנדרשת היא כנגד יריב כל יכול אך הגנה זו ניתנת להוכחה באופן שלא מסתמך על הנחות לבין קריפטוגרפיה מבוססת חישוביות בה ההגנה הנדרשת היא כנגד יריבים מוגבלים חישובית אך מסתמכת על הנחות קושי בלתי מוכחות.

פרק 4 מציג בנייה של סכמת שיתוף סוד מתוך קוד נתון ואת הקשרים בין הפרמטרים של הקוד לאלו של הסכמה המתקבלת. בעזרת בנייה זו יחד עם הקודים שלנו אנו מקבלים סכימת כמעט־סף לשיתוף סוד בה שיתוף הסוד ושיחזורו ניתנים לביצוע בזמן לינארי, תוצאה ראשונה מסוג זה. בנייה זו מסתמכת על העבודה שהקודים שלנו (בדומה לקודים אקראיים) משיגים לא רק מרחק טוב אלא גם מרחק דואלי טוב.

פרק 5 עוסק באפליקציות קריפטוגרפיות מבוססות סיבוכיות של הקודים שלנו. הדיון מתחיל ברדוקצית חיפוש להכרעה (התקפה גם למשפחות קודים כלליות). הנחת הקושי של הקודים שלנו מוצגת בהמשך כאשר הרדוקציה לעיל שמה אותנו במצב מועדף בו גם הפרכה של הנחה זו משמעותה תוצאות חדשות. אנו מציגים בניות גנריות של הצפנה סימטרית וסכימת זיהוי מבוססות קודים כאשר בטיחות הבניות נובעת ישירות מקושי ההכרעה של הקוד המדובר. על ידי שימוש בהנחת הקושי של הקודים שלנו אנו מקבלים מועמדים לבניות בעלות יחס יעילות/בטיחות טוב יותר מבניות הידועות כיום.

פרק 6 מסכם את התוצאות ומעלה כמה שאלות פתוחות הנוגעות להן.