Improved Trade-Offs Between Amortization and Download Bandwidth for Linear HSS

Keller Blackwell
Department of Computer Science, Stanford University, CA, USA

Mary Wootters
Departments of Computer Science and Electrical Engineering, Stanford University, CA, USA

Abstract

A *Homomorphic Secret Sharing* (HSS) scheme is a secret-sharing scheme that shares a secret x among s servers, and additionally allows an output client to reconstruct some function $f(x)$ using information that can be locally computed by each server. A key parameter in HSS schemes is *download rate*, which quantifies how much information the output client needs to download from the servers. Often, download rate is improved by *amortizing* over ℓ instances of the problem, making ℓ also a key parameter of interest.

Recent work [23] established a limit on the download rate of linear HSS schemes for computing low-degree polynomials and constructed schemes that achieve this optimal download rate; their schemes required amortization over $\ell = \Omega(s \log(s))$ instances of the problem. Subsequent work [6] completely characterized linear HSS schemes that achieve optimal download rate in terms of a coding-theoretic notion termed *optimal labelweight codes*. A consequence of this characterization was that $\ell = \Omega(s \log(s))$ is in fact necessary to achieve optimal download rate.

In this paper, we characterize *all* linear HSS schemes, showing that schemes of any download rate are equivalent to a generalization of optimal labelweight codes. This equivalence is constructive and provides a way to obtain an explicit linear HSS scheme from *any* linear code. Using this characterization, we present explicit linear HSS schemes with slightly sub-optimal rate but with much improved amortization $\ell = O(s)$. Our constructions are based on algebraic geometry codes (specifically Hermitian codes and Goppa codes).

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Error-correcting codes

Keywords and phrases Error Correcting Codes, Homomorphic Secret Sharing

Digital Object Identifier 10.4230/LIPIcs.ITC.2024.7

Related Version *Full Version*: `https://arxiv.org/abs/2403.08719`

Funding *Keller Blackwell*: KB is supported by a National Science Foundation Graduate Research Fellowship and by a graduate fellowship award from Knight-Hennessy Scholars at Stanford University. KB's work is supported partially supported by NSF grant CCF-2231157.
Mary Wootters: MW's work was partially supported by NSF grants CCF-1844628, CCF-2133154, and CCF-2231157.

Acknowledgements We thank the anonymous referees for helpful feedback.

1 Introduction

A *Homomorphic Secret Sharing* (HSS) scheme is a secret sharing scheme that supports computation on top of the shares [4, 12, 13]. Homomorphic Secret Sharing has been a useful primitive in cryptography, with applications ranging from private information retrieval to secure multiparty computation (see, e.g., [9, 13]).

In this work, we focus on information-theoretically secure HSS schemes for the class of degree d, m-variate polynomials. Suppose m secrets, $x_1, \ldots, x_m$, are shared independently with a t-private secret sharing scheme Share[1]and that server j receives the m shares $y_{k,j}$ for $k \in [m]$. Denote by $\mathrm{POLY}_{d,m}(\mathbb{F}) \subseteq \mathbb{F}[X_1, \ldots, X_m]$ an arbitrary set of degree d, m-variate polynomials. Given a polynomial $f \in \mathrm{POLY}_{d,m}(\mathbb{F})$, each server j does some local computation on its shares $y_{k,j}$ for $k \in [m]$ to obtain an *output share* $z_j = \mathsf{Eval}(f, j, (y_{1,j}, \ldots, y_{m,j}))$. An *output client* receives the output shares $z_1, \ldots, z_s$ and runs a recovery algorithm Rec to obtain $f(x_1, \ldots, x_m) = \mathsf{Rec}(z_1, \ldots, z_s)$. The HSS scheme π is given by the tuple of functions $(\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$; see Definition 7 for a formal definition.

Parameters of interest

A key parameter of interest in an HSS scheme is the *download rate* (Definition 12), which is the ratio of the number of bits in $f(x)$ to the number of bits in all of the output shares z_j. Ideally this rate would be as close to 1 as possible, because that would mean that the output client does not have to download too much more information than it wishes to compute.

Another parameter of interest is the *amortization* that the scheme uses. As in previous work [23, 6], we consider HSS schemes for low-degree polynomials that amortize over ℓ instances of the problem. This means that we have ℓ batches of m secrets, $x_k^{(i)}$ for $i \in [\ell]$ and $k \in [m]$, and ℓ polynomials $f_1, \ldots, f_\ell$. Each of these $m\ell$ secrets is shared independently, as before, but the output shares z_j are allowed to depend on *all* ℓ batches. Then the output client is responsible for computing $f_i(x_1^{(i)}, \ldots, x_m^{(i)})$ for all $i \in [\ell]$.

Trade-offs between download rate and amortization

[23, 6] previously studied the optimal download rate possible for linear HSS schemes[2], and also studied what amount of amortization is necessary to obtain this optimal rate. In this work, we show that by backing off from the optimal rate by a small amount, one can get asymptotic improvements in the amortization parameter.

In more detail, [23] showed that for t-private, s-server linear HSS schemes for m-variate degree-d polynomials, the best download rate possible is $1 - dt/s$. They achieved this download rate with schemes that had amortization $\ell = \Omega(s \log(s))$. In follow-up work, [6] showed that in fact amortization $\ell = \Omega(s \log(s))$ was *necessary* to achieve the optimal download rate of $1 - dt/s$. Their result followed from a characterization optimal-rate linear HSS schemes in terms of a coding theoretic notion they introduce, termed *optimal labelweight codes*.

In our work, informally, we show that by backing off from the optimal rate by a small amount, we are able to get asymptotic improvements in the amortization required; in some cases we need only $\ell = O(s)$. We obtain this by generalizing the characterization from [6] to *all* HSS schemes, not just optimal ones. We discuss our main results below in more detail.

---

[1] A t-private secret sharing scheme shares a secret x among s servers by computing s *shares*, $\mathsf{Share}(x) = (y_1, \ldots, y_s)$. The t-privacy guarantee means that no t of the servers should be able to learn anything about x given their shares.

[2] A *linear* HSS scheme is a scheme where both Share and Rec are linear over some field $\mathbb{F}$. Note that Eval need not be linear.

1.1 Main Results

For all of our results, we consider *CNF sharing* [27] (see Definition 16). It is known that CNF sharing is universal for linear secret sharing schemes, in that t-CNF shares can be locally converted to shares of *any* linear t-private secret sharing scheme [18].

Main Contributions

1. **A complete characterization of the Rec functions for *all* linear HSS schemes for $\mathbf{POLY}_{d,m}(\mathbb{F})$.** As mentioned above, [6] gives a characterization of *optimal*-download-rate linear HSS schemes in terms of codes with good labelweight. In our work, we extend that characterization to *all* linear HSS schemes.
   Our characterization is constructive, and in particular it gives an efficient algorithm to convert any code with good labelweight into a linear HSS scheme, and vice-versa.
2. **Improved amortization without much loss in rate.** The work [6] showed that to achieve optimal rate, it was necessary to have amortization $\ell = \Omega(s \log s)$. Leveraging our characterization from Item 1, we give efficient constructions of linear HSS schemes that achieve *near*-optimal download rate while requiring amortization parameter only $\ell = O(s)$. We compute the parameters of our constructions in practical parameter regimes and show that our schemes achieve a near order-of-magnitude savings in amortization parameter, even for reasonable values of s, d, m.

We describe our results in greater detail below.

(1) Characterization of arbitrary-rate linear HSS schemes

Theorem 2 below is a characterization of *all* linear HSS schemes for $\mathrm{POLY}_{d,m}(\mathbb{F})$. In particular, our characterization extends that of [6], which only characterized optimal-rate linear HSS schemes. We show that the Rec algorithms for such schemes (with CNF sharing) are equivalent to a class of linear codes with sufficiently good *labelweight*, a generalization of Hamming distance that was introduced by [6].

▶ **Definition 1** (Labelweight). *Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code of dimension ℓ. Let $\mathcal{L} : [n] \to [s]$ be any surjective function, which we refer to as a* labeling *function. The* labelweight *of $\mathbf{c} \in \mathcal{C}$ is the number of distinct labels that the support of $\mathbf{c}$ touches:*

$$\Delta_{\mathcal{L}}(\mathbf{c}) = |\{\mathcal{L}(i) \,:\, i \in [n], c_i \neq 0\}|.$$

The labelweight *of $\mathcal{C}$ is the minimum labelweight of any nonzero codeword:*

$$\Delta_{\mathcal{L}}(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \Delta_{\mathcal{L}}(\mathbf{c}).$$

In particular, if $s = n$ and $\mathcal{L}(j) = j$ for all $j \in [n]$, then $\Delta_{\mathcal{L}}(\mathcal{C})$ is just the minimum Hamming distance of $\mathcal{C}$. Thus, the labelweight of a code generalizes the standard notion of distance.

Our main characterization theorem is the following.

▶ **Theorem 2** (Linear HSS schemes are equivalent to labelweight codes. (Informal, see Theorem 18)). *Let $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ be a t-private, s-server linear HSS for $\mathrm{POLY}_{d,m}(\mathbb{F})$ with download rate R and amortization parameter ℓ. Let $G \in \mathbb{F}^{\ell \times (\ell/R)}$ be the matrix that represents* Rec *(see Observation 10). Then there is some labeling function $\mathcal{L}$ so that G is the generator matrix for a code $\mathcal{C}$ of dimension ℓ, with rate R and with $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$.*

Conversely, suppose that there is a labeling function $\mathcal{L} : [n] \to [s]$ *and a linear code* $\mathcal{C} \subseteq \mathbb{F}^n$ *of dimension* ℓ *with rate* R *and* $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$*. Then any generator matrix* G *of* $\mathcal{C}$ *describes a linear reconstruction algorithm* Rec *for an* s*-server* t*-private linear HSS for* $\mathrm{POLY}_{d,m}(\mathbb{F})$ *that has download rate* R *and amortization parameter* ℓ*.*

We remark that the converse direction is constructive: given the description of such a code $\mathcal{C}$, the proof (see Theorem 20) gives an efficient construction of the Eval function as well as the Rec function.

(2) Achieving practical trade-off between download rate and amortization parameter

Using the complete characterization of all linear HSS schemes, we construct linear HSS schemes that achieve near-optimal download rate at amortization parameters that are strictly linear in in the number of servers s. Through the lens of Theorem 2, this is equivalent to constructing high-labelweight, high-rate linear codes.

While the construction of [6] use Reed-Solomon codes as a starting point to constructing optimal rate linear HSS schemes, we use two well-studied families of algebraic geometry (AG) codes – Hermitian codes and Goppa codes. For any code $\mathcal{C}$, observe that the minimum labelweight $\Delta_{\mathcal{L}}(\mathcal{C})$ is sharply upper-bounded by the code's minimum distance. Therefore, so as to maximize labelweight, we use the trivial labeling scheme where $n = s$ and $\mathcal{L} : [n] \to [s], x \mapsto x$ is the identity function. Such labelweight codes, though straightforward, yield linear HSS schemes with attractive parameters for realistic server counts. Furthermore, their intuitive construction underscores the fundamental relationship between classical codes and linear HSS. We loosely summarize these results in Table 1; see Theorems 22, 27 for additional details and formal statements.

Table 1 Comparing our AG-based constructions to [23], [6].

	[23],[6]	Hermitian-based HSS	Goppa-based HSS
Download Rate	$1 - dt/s$	$1 - dt/s - O(s^{-1/3})$	$1 - (dt/s) \cdot O(\log(dt))$
Amortization	$(s - dt)\log(s)$	$s - dt - O(s^{2/3})$	$s - dt \cdot O(\log(dt))$

Furthermore, for completeness we show in Appendix A that, perhaps surprisingly, a random coding approach does not lead to amortization savings over [23], [6], even backing off from the optimal rate. More precisely, linear HSS schemes instantiated from random labelweight codes result in HSS schemes with amortization parameter at least $\Omega(s \log(s))$. This motivates additional study of linear codes with algebraic structure as a basis for linear HSS with attractive parameters.

1.2 Technical Overview

In this section, we give a high-level overview of the techniques underpinning Theorem 2, which states that any t-private, s-server linear HSS scheme for $\mathrm{POLY}_{d,m}(\mathbb{F})$ is equivalent to a labelweight code with minimum labelweight $\geq dt + 1$.

To give some intuition for the connection, we recount the simplest non-trivial case of the forward direction, which was proven by [6]. We consider *HSS for concatenation* [23]: ℓ secrets $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(\ell)}\right) \in \mathbb{F}^\ell$ are shared independently among s servers who in turn communicate them to an output client. The objective is for the output client to download as little information as possible - in particular, significantly less than the naive solution of simply downloading $t + 1$ shares of each secret.

Let $\mathbf{z} \in \mathbb{F}^n$ be the n-tuple of $\mathbb{F}$-symbols downloaded by the output client; since the output client instantiates a linear reconstruction algorithm Rec, there exists $G \in \mathbb{F}^{\ell \times n}$ such that $G\mathbf{z} = \mathbf{x}$. Define a labeling function $L : [n] \to [s]$ satisfying the property that for all $i \in [n]$, $L(i) = r \in [s]$ if and only if $\mathbf{z}_i$ was downloaded from server r.

▷ Claim 3. The rows of G generate a linear code $\mathcal{C}$ satisfying $\Delta_{\mathcal{L}}(\mathcal{C}) \geq t + 1$.

Proof. Suppose towards a contradiction that for some non-zero $\mathbf{m} \in \mathbb{F}^\ell$, $\Delta_{\mathcal{L}}(\mathbf{m}G) \leq t$. Then $\mathbf{m}G\mathbf{x} = \mathbf{c}\mathbf{x}$ (for some non-zero $\mathbf{c} \in \mathbb{F}^n$) is a linear combination of secrets recoverable by a set of t servers, contradicting the t-privacy they were originally secret shared with. ◁

In this example, the function evaluated on the secret shares is identity; the generalization of this example requires consideration of more general functions, but the fundamental principle is similar.

The converse requires showing that any labelweight code $\mathcal{C}$ implies a linear HSS scheme. In the setting of optimal download rate, [6] leveraged the specific properties of optimal labelweight codes in order to prove this; their work relied on the fact that optimal labelweight codes are highly structured. In particular, [6] showed that, in the optimal download rate setting:

(i) the output client must download an equal number of symbols from each server; and
(ii) up to elementary row operations, the matrix parameterizing the output client's linear Rec algorithm is a rectangular array of invertible matrices, with the property that any square sub-array is itself invertible.

These strong symmetry properties are key to the equivalence result of [6]. However, in our setting, where we do not assume that the optimal download rate is attained, *neither of the aforementioned properties hold.* Our result thus requires proving additional properties of labelweight codes.

1.3 Related Work

Though linear HSS schemes are implicit in classical protocols for secure multi-party computation and private information retrieval [4, 3, 15, 19, 1, 2, 16], the systematic study of HSS was introduced by [13]. Most HSS schemes reply on cryptographic hardness assumptions [10, 21, 11, 12, 22, 13, 14, 8, 17, 28, 29, 20].

In contrast, the HSS schemes presented in this work are *information-theoretically secure.* The information-theoretic setting was explored in [13] and was further studied in [23] and [6]; these latter two works are the closest to our work, and we discuss them more below.

The work of [23] focused on the download rate of information-theoretically secure HSS schemes (both linear and non-linear) and proved a tight impossibility result regarding the highest download rate achievable by linear HSS schemes. They paired this with an explicit construction that showed a large amortization parameter is a sufficient condition for a linear HSS scheme to achieve optimal download rate.

▶ **Theorem 4** ([23]). *Let $s, d, t \in \mathbb{Z}^+$ such that $s > dt$. Let π be a t-private, s-server linear HSS scheme for* $\mathrm{POLY}_{d,m}(\mathbb{F})$. *Then* $\mathsf{DownloadRate}(\pi) \leq 1 - dt/s$.

Furthermore, for all integers $j \geq \log_{|\mathbb{F}|}(s)$, there exists a t-private, s-server linear HSS π satisfying $\mathsf{DownloadRate}(\pi) = 1 - dt/s$ *with amortization parameter $\ell = j(s - dt)$.*

The work [6] focused on linear schemes with *optimal* download rate, meeting the bound showed in [23]. They proved that in fact a large amortization parameter is necessary for a linear HSS scheme to achieve optimal download rate.

▶ **Theorem 5** ([6]). *There exists a t-private, s-server linear HSS scheme for $POLY_{d,m}(\mathbb{F})$ with download rate $(s-dt)/s$ and amortization parameter ℓ only if $\ell = j(s-dt)$ for some $j \in \mathbb{Z}+$ satisfying*

$$j \geq \lceil \max\left\{\log_q(s-dt+1), \log_q(dt+1)\right\}\rceil.$$

The key technique in their proof was showing that optimal-rate linear HSS is in fact equivalent to optimal labelweight linear codes; more precisely, they showed the following theorem.

▶ **Theorem 6** ([6]). *There exists a t-private, s-server linear HSS scheme for $POLY_{d,m}(\mathbb{F})$ with download rate $(s-dt)/s$ and amortization parameter ℓ if and only if there exists a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with information rate $(s-dt)/s$ and dimension ℓ; and surjection $L : [n] \to [s]$ such that $\Delta_L(\mathcal{C}) \geq dt+1$.*

Our work extends the characterization of [6] to *all* linear HSS schemes with arbitrary download rate; in particular, we show that arbitrary-rate linear HSS schemes are equivalent to a broader class of labelweight codes than those considered by [6]. Though our proof syntactically resembles that of Theorem 6 from [6], as discussed in Section 1.2, we need to overcome additional technical difficulties introduced by the lack of strong symmetries in the more general arbitrary-rate setting. Furthermore, we present explicit constructions that approach the optimal download rate from Theorem 15, while asymptotically improving the amortization parameter ℓ.

1.4 Organization

In Section 2, we set notation and record a few formal definitions that we will need. In Section 3, we show that linear HSS schemes (with arbitrary download rate) are equivalent to codes with sufficient labelweight: Lemma 19 establishes that HSS schemes imply codes with sufficient labelweight, and Theorem 20 constructively establishes the converse.

In Section 4, we derive labelweight codes from Hermitian codes and construct the corresponding linear HSS scheme by Theorem 20. We formally state its parameters in Theorem 22 and compare its performance against constructions from [23] and [6]. In Section 5, we do the same but use Goppa codes as the basis for a family of labelweight codes.

2 Preliminaries

We begin by setting notation and the basic definitions that we will need throughout the paper. (We note that these definitions and notation closely follow that of [23] and [6]).

Notation. For $n \in \mathbb{Z}^+$, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. We use bold symbols (e.g., $\mathbf{x}$) to denote vectors. For an object w in some domain $\mathcal{W}$, we use $\|w\| = \log_2(|\mathcal{W}|)$ to denote the number of bits used to represent w.

2.1 Homomorphic Secret Sharing

We consider homomorphic secret sharing (HSS) schemes with m inputs and s servers; each input is shared independently. We denote by $\mathcal{F} = \{f : \mathcal{X}^m \to \mathcal{O}\}$ the class of functions we wish to compute, where $\mathcal{X}$ and $\mathcal{O}$ are input and output domains, respectively.

▶ **Definition 7** (HSS). *Given a collection of s servers and a function class $\mathcal{F} = \{f : \mathcal{X}^m \to \mathcal{O}\}$, consider a tuple $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$, where $\mathsf{Share} : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}^s$, $\mathsf{Eval} : \mathcal{F} \times [s] \times \mathcal{Y} \to \mathcal{Z}^*$, and $\mathsf{Rec} : \mathcal{Z}^* \to \mathcal{O}$ are as follows*[3]*:*

- $\mathsf{Share}(x_i, r_i)$*: For $i \in [m]$,* Share *takes as input a secret $x_i \in \mathcal{X}$ and randomness $r_i \in \mathcal{R}$; it outputs s shares $(y_{i,j} : j \in [s]) \in \mathcal{Y}^s$. We refer to the $y_{i,j}$ as* input shares*; server j holds shares $(y_{i,j} : i \in [m])$.*
- $\mathsf{Eval}\,(f, j, (y_{1,j}, y_{2,j}, \ldots, y_{m,j}))$*: Given $f \in \mathcal{F}$, server index $j \in [s]$, and server j's input shares $(y_{1,j}, y_{2,j}, \ldots, y_{m,j})$,* Eval *outputs $z_j \in \mathcal{Z}^{n_j}$, for some $n_j \in \mathbb{Z}$. We refer to the z_j as* output shares*.*
- $\mathsf{Rec}(z_1, \ldots, z_s)$*: Given output shares $z_1, \ldots, z_s$,* Rec *computes $f(x_1, \ldots, x_m) \in \mathcal{O}$.*

We say that $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ is a s-server HSS scheme for $\mathcal{F}$ if the following requirements hold:

- ***Correctness:*** *For any m inputs $x_1, \ldots, x_m \in \mathcal{X}$ and $f \in \mathcal{F}$,*

$$\Pr_{\mathbf{r} \in \mathcal{R}^m}\left[\mathsf{Rec}(z_1, \ldots, z_s) = f(x_1, \ldots, x_m) : \begin{array}{l} \forall i \in [m],\ (y_{i,1}, \ldots, y_{i,s}) \leftarrow \mathsf{Share}(x_i, r_i) \\ \forall j \in [s],\ z_j \leftarrow \mathsf{Eval}\,(f, j, (y_{1,j}, \ldots, y_{m,j})) \end{array}\right] = 1$$

  Note that the random seeds $r_1, \ldots, r_m$ are independent.
- ***Security:*** *Fix $i \in [m]$; we say that π is* t-private *if for every $T \subseteq [s]$ with $|T| \leq t$ and $x_i, x_i' \in \mathcal{X}$, $\mathsf{Share}(x_i)|_T$ has the same distribution as $\mathsf{Share}(x_i')|_T$, over the randomness $\mathbf{r} \in \mathcal{R}^m$ used in* Share*.*

▶ Remark 8. We remark that in the definition of HSS, the reconstruction algorithm Rec does *not* need to know the identity of the function f being computed, while the Eval function does. In some contexts it makes sense to consider an HSS scheme for $\mathcal{F} = \{f\}$, in which case f is fixed and known to all. Our results in this work apply for general collections $\mathcal{F}$ of low-degree, multivariate polynomials, and in particular cover both situations.

We focus on *linear* HSS schemes, where both Share and Rec are $\mathbb{F}$-linear over some finite field $\mathbb{F}$; note that Eval need not be linear.

▶ **Definition 9** (Linear HSS). *Let $\mathbb{F}$ be a finite field.*

- *We say that an s-server HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ has* linear reconstruction *if:*
  - *$\mathcal{Z} = \mathbb{F}$, so each output share $z_i \in \mathbb{F}^{n_i}$ is a vector over $\mathbb{F}$;*
  - *$\mathcal{O} = \mathbb{F}^o$ is a vector space over $\mathbb{F}$; and*
  - *$\mathsf{Rec} : \mathbb{F}^{\sum_i n_i} \to \mathbb{F}^o$ is $\mathbb{F}$-linear.*
- *We say that π has* linear sharing *if $\mathcal{X}$, $\mathcal{R}$, and $\mathcal{Y}$ are all $\mathbb{F}$-vector spaces, and* Share *is $\mathbb{F}$-linear.*
- *We say that π is* linear *if it has both linear reconstruction and linear sharing. Note there is no requirement for* Eval *to be $\mathbb{F}$-linear.*

The assumption of linearity implies that the function Rec can be represented by a matrix, as per the following observation that was also used by [6].

▶ **Observation 10** ([6]). *Let ℓ, t, s, d, m, n be integers. Let $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ be a t-private, s-server HSS for some function class $\mathcal{F} \subseteq \mathrm{POLY}_{d,m}(\mathbb{F})^\ell$ with linear reconstruction $\mathsf{Rec} : \mathbb{F}^n \to \mathbb{F}^\ell$.*

[3] By $\mathcal{Z}^*$, we mean a vector of some number of symbols from $\mathcal{Z}$.

Then there exists a matrix $G_\pi \in \mathbb{F}^{\ell \times n}$ so that, for all $f \in \mathcal{F}$ and for all secrets $\mathbf{x} \in (\mathbb{F}^m)^\ell$, there exists some $\mathbf{z} \in \mathcal{F}^n$ such that

$$\mathsf{Rec}(\mathbf{z}) = G_\pi \mathbf{z} = f(\mathbf{x}) = \left[f_1(\mathbf{x}^{(1)}),\ f_2(\mathbf{x}^{(2)}), \ldots,\ f_\ell(\mathbf{x}^{(\ell)})\right]^\mathsf{T}.$$

For a linear HSS π, we call G_π as in the observation above the *reconstruction matrix* corresponding to Rec. It was shown in [6] that any such reconstruction matrix must be full rank.

▶ **Lemma 11** ([6]). *Let t, s, d, m, ℓ be positive integers so that $m \geq d$ and $n \geq \ell$, and let π be a t-private s-server linear HSS for some $\mathcal{F} \subseteq \mathrm{POLY}_{d,m}(\mathbb{F})$, so that $\mathcal{F}$ contains an element $(f_1, \ldots, f_\ell)$ where for each $i \in [\ell]$, f_i is non-constant. Then $G_\pi \in \mathbb{F}^{\ell \times n}$ has rank ℓ.*

Finally, we formally define the *download rate* of an HSS scheme.

▶ **Definition 12** (Download cost, dowload rate). *Let s, t be integers and let $\mathcal{F}$ be a class of functions with input space $\mathcal{X}^m$ and output space $\mathcal{O}$. Let π be an s-server t-private HSS for $\mathcal{F}$. Let $z_i \in \mathcal{Z}^{n_i}$ for $i \in [s]$ denote the output shares.*

- *The* download cost *of π is given by*

$$\mathsf{DownloadCost}(\pi) := \sum_{i \in [s]} \|z_i\|,$$

  where we recall that $\|z_i\| = n_i \log_2 |\mathcal{Z}|$ denotes the number of bits used to represent z_i.
- *The* download rate *of π is given by*

$$\mathsf{DownloadRate}(\pi) := \frac{\log_2 |\mathcal{O}|}{\mathsf{DownloadCost}(\pi)}.$$

Thus, the download rate is a number between 0 and 1, and we would like it to be as close to 1 as possible.

2.2 Polynomial Function Classes

Throughout, we will be interested in classes of functions $\mathcal{F}$ comprised of low-degree polynomials.

▶ **Definition 13.** *Let $m > 0$ be an integer and $\mathbb{F}$ be a finite field. We define*

$$\mathrm{POLY}_{d,m}(\mathbb{F}) := \{f \in \mathbb{F}[X_1, \ldots, X_m] : \deg(f) \leq d\}$$

to be the class of all m-variate polynomials of degree at most d, with coefficients in $\mathbb{F}$.

We are primarily interested in *amortizing* HSS computation over ℓ instances of $\mathrm{POLY}_{d,m}(\mathbb{F})$, as discussed in the Introduction. We can capture this as part of Definition 7 by taking the function class $\mathcal{F}$ to be (a subset of) $\mathrm{POLY}_{d,m}(\mathbb{F})^\ell$ for some $\ell \in \mathbb{Z}^+$. Note that this corresponds to the amortized setting discussed in the Introduction.

▶ **Definition 14.** *Let $\mathcal{F} \subseteq \mathrm{POLY}_{d,m}(\mathbb{F})^\ell$. We say that $\mathcal{F}$ is* non-trivial *if there exists some $\mathbf{f} = (f_1, \ldots, f_\ell) \in \mathcal{F}$ so that for all $i \in [\ell]$, f_i contains a monomial with at least d distinct variables.*

The work [23] showed that any linear HSS scheme for $\mathrm{POLY}_{d,m}(\mathbb{F})^\ell$ (for any ℓ) can have download rate at most $(s - dt)/s$: We recall the following theorem from [23].

▶ **Theorem 15** ([23]). *Let t, s, d, m, ℓ be positive integers so that $m \geq d$. Let $\mathbb{F}$ be any finite field and π be a t-private s-server linear HSS scheme for $\mathrm{POLY}_{d,m}(\mathbb{F})^\ell$. Then $dt < s$, and $\mathsf{DownloadRate}(\pi) \leq (s - dt)/s$.*

2.3 CNF Sharing

The main Share function that we consider in this work is *CNF sharing* [27].

▶ **Definition 16** (t-private CNF sharing). *Let $\mathbb{F}$ be a finite field. The t-private, s-server CNF secret-sharing scheme over $\mathbb{F}$ is a function* $\mathsf{Share} : \mathbb{F} \times \mathbb{F}^{\binom{s}{t}-1} \to \left(\mathbb{F}^{\binom{s-1}{t}}\right)^s$ *that shares a secret $x \in \mathbb{F}$ as s shares $y_j \in \mathbb{F}^{\binom{s-1}{t}}$, using $\binom{s}{t} - 1$ random field elements, as follows.*

Let $x \in \mathbb{F}$, and let $\mathbf{r} \in \mathbb{F}^{\binom{s}{t}-1}$ be a uniformly random vector. Using $\mathbf{r}$, choose $y_T \in \mathbb{F}$ for each set $T \subseteq [s]$ of size t, as follows: The y_T are uniformly random subject to the equation

$$x = \sum_{T \subseteq [s] : |T| = t} y_T.$$

Then for all $j \in [s]$, define $\mathsf{Share}(x, \mathbf{r})_j = (y_T \,:\, j \notin T) \in \mathbb{F}^{\binom{s-1}{t}}$.

We observe that CNF-sharing is indeed t-private. Any $t+1$ servers between them hold all of the shares y_T, and thus can reconstruct $x = \sum_T y_T$. In contrast, any t of the servers (say given by some set $S \subseteq [s]$) are missing the share y_S, and thus cannot learn anything about x.

The main reason we focus on CNF sharing is that it is *universal* for linear secret sharing schemes:

▶ **Theorem 17** ([18]). *Suppose that $x \in \mathbb{F}$ is t-CNF-shared among s servers, so that server j holds $y_j \in \mathbb{F}^{\binom{s-1}{t}}$, and let* Share' *be any other linear secret-sharing scheme for s servers that is (at least) t-private. Then the shares y_j are locally convertible into shares of* Share'. *That, is there are functions $\phi_1, \ldots, \phi_s$ so that $(\phi_1(y_1), \ldots, \phi_s(y_s))$ has the same distribution as* $\mathsf{Share}'(x, \mathbf{r})$ *for a uniformly random vector $\mathbf{r}$.*

2.4 Linear Codes

Throughout, we will be working with *linear codes* $\mathcal{C} \subset \mathbb{F}^n$, which are just subspaces of $\mathbb{F}^n$. For a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ of dimension ℓ, a matrix $G \in \mathbb{F}^{\ell \times n}$ is a *generator matrix* for $\mathcal{C}$ if $\mathcal{C} = \mathrm{rowSpan}(G)$. Note that generator matrices are not unique. The *rate* of a linear code $\mathcal{C} \subset \mathbb{F}^n$ of dimension ℓ is defined as $\mathsf{Rate}(\mathcal{C}) := \frac{\ell}{n}$.

3 Equivalence of Linear HSS and Labelweight Codes

In this section we show that linear HSS schemes for low-degree multivariate polynomials are equivalent to linear codes with sufficient labelweight. Concretely, we have the following theorem, which formalizes the statement of Theorem 2.

▶ **Theorem 18.** *Let ℓ, t, s, d, m, n be integers, with $m \geq d$, $\ell \leq n$. There exists a t-private, s-server $\mathbb{F}$-linear HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ for any non-trivial $\mathcal{F} \subseteq POLY_{d,m}(\mathbb{F})^\ell$, with download rate $\mathsf{DownloadRate}(\pi) = \ell/n$, if and only if there exists a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with rate $\mathsf{DownloadRate}(\pi)$ and a labeling $\mathcal{L} : [n] \to [s]$ so that $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$.*

The work of [6] proved this equivalence for only the optimal-rate setting and left the equivalence in an arbitrary-rate setting as an open question. Theorem 18 settles this question and shows that linear HSS and linear codes of sufficient labelweight are indeed equivalent in *all* parameter regimes. The proof of Theorem 18 follows from Lemma 19 (for the forward direction) and Theorem 20 (for the converse) below.

We begin with the forward direction.

▶ **Lemma 19** (Follows from the analysis of [6]). *Let ℓ, t, s, d, m, n be integers, with $m \geq d$, $\ell \leq n$. Suppose there exists a t-private, s-server $\mathbb{F}$-linear HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ for any non-trivial (see Definition 14) $\mathcal{F} \subseteq POLY_{d,m}(\mathbb{F})^{\ell}$, with download rate $\mathsf{DownloadRate}(\pi) = \ell/n$. Then there exists a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with rate $\mathsf{DownloadRate}(\pi)$ and a labeling $\mathcal{L} : [n] \to [s]$ so that $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$.*

Though the statement of Lemma 19 is more general than its optimal-rate counterpart in [6], the proof is analogous; a careful reading of Lemma 12 in [6] shows that this forward direction does not leverage any of the strong symmetries of optimal rate linear HSS. Thus, we refer the reader to [6] for a proof.

Thus, in the rest of this section we focus on the converse, which does deviate from the analysis of [6], as we cannot leverage the same strong symmetries that they did, as discussed in Section 1.2. We first formally state the converse.

▶ **Theorem 20.** *Let ℓ, t, s, d, m, n be integers, with $m \geq d$. Suppose that there exists a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with dimension ℓ and rate ℓ/n. Suppose there exists a labeling $\mathcal{L} : [n] \to [s]$ so that $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$. Then there exists a t-private, s-server linear HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ for $\mathrm{POLY}_{d,m}(\mathbb{F})^{\ell}$ with download rate ℓ/n and amortization parameter ℓ.*

The main ingredient in proving this direction without the strong symmetries of optimal rate linear HSS is the following lemma, which neatly generalizes the results of Lemma 13 and Corollaries 14, 15 of [6].

▶ **Lemma 21.** *Let $\mathcal{C}$ be a length n, dimension ℓ linear code over $\mathbb{F}_q$ with generator matrix $G \in \mathbb{F}_q^{\ell \times n}$. Let $\mathcal{L} : [n] \to [s]$ be a surjective labeling such that $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$.*

For $\Lambda \subseteq [s]$, let $G(\Lambda)$ denote the restriction of G to the columns $r \in [n]$ so that $\mathcal{L}(r) \in \Lambda$. Then for any $|\Lambda| \geq s - dt$, $G(\Lambda)$ has full row rank.

Proof. Let $\Lambda = \Lambda' \cup \Lambda''$ where $\Lambda' \cap \Lambda'' = \varnothing$ and $|\Lambda'| = s - dt$. If $G(\Lambda')$ achieves full row rank, then so does $G(\Lambda)$, since adding columns to a matrix does not induce linear independence among its rows. Hence, it suffices to consider only $|\Lambda| = s - dt$.

Up to a permutation of columns, G can be written as $G = [\, G(\Lambda) \mid G([s] \setminus \Lambda)]$. Let w denote the number of columns in $G(\Lambda)$.

Assume towards a contradiction that there exists some $\mathbf{v} \in \mathbb{F}_q^{\ell}$ such that $\mathbf{v}G(\Lambda) = 0^w$. Then

$$\mathbf{v}G = [\mathbf{v}G(\Lambda) \mid \mathbf{v}G([s] \setminus \Lambda)] = [0^w \mid \mathbf{v}G([s] \setminus \Lambda)]\,.$$

Since $|[s] \setminus \Lambda| = dt$, it follows that

$$\Delta_{\mathcal{L}}(\mathbf{v}G) = \Delta_{\mathcal{L}}(\mathbf{v}G([s] \setminus \Lambda)) \leq dt$$

which contradicts $\Delta_{\mathcal{L}}(\mathcal{C}) \geq dt + 1$. ◀

Let G_π be a reconstruction matrix. At a high level, Lemma 21 says that any sufficiently large submatrix of G_π, obtained by only considering columns labeled with a sufficiently large subset $\Lambda \subseteq [s]$, must be full-rank. [6] proved that such a property held for G_π in the optimal-rate regime that relied heavily on the fact that, in optimal-rate linear HSS, the output client downloads an equal number of output symbols from each server. This is equivalent to requiring that the sets $\mathcal{L}^{-1}(y) := \{x \in [n] : \mathcal{L}(x) = y\}$ be the same size for all $y \in [s]$. The proof of Lemma 21 shows that, perhaps surprisingly, sufficiently large submatrices of G_π still achieve full-rank even when the output client is allowed to download arbitrary numbers of output symbols from each server.

The remainder of the proof of Theorem 20 proceeds in a familiar syntax to that of [6]; we omit its presentation here and refer the interested reader to the full manuscript [7].

4 Linear HSS from Hermitian Codes

Linear HSS schemes presented by [23], [6] both achieve optimal download rate $1 - dt/s$ but require large amortization parameters to do so. [23] showed it was sufficient to take amortization parameter $\ell = (s - dt)\log(s) = O(s\log(s))$, and [6] proved that such an amortization parameter is in fact necessary in many parameter regimes, and is off by at most 1 otherwise.

It is a natural to ask whether linear HSS schemes that achieve a better trade-off between download rate and amortization parameter exist. *Specifically, can we make minor concessions to download rate and save substantially on the amortization needed?*

Through the lens of Theorem 20, this is equivalent to asking whether there exists a labelweight code with minimum labelweight $\geq dt + 1$ that achieves good rate at low dimension. A natural first attempt at an existential result would be via random coding; specifically, building a linear HSS scheme by starting with a random linear code and following the construction of Theorem 20. Unfortunately (and perhaps surprisingly!), we show in Appendix A that this results in strictly worse parameters than [23], [6].

In the following sections we take a different approach. We derive our labelweight codes straightforwardly from well-studied algebraic geometric codes: we set the number of servers s equal to the block length n of the codes and label each coordinate by the identity function $\mathcal{L} : [n] \to [s], x \mapsto x$. In this setting, labelweight is equivalent to Hamming weight. Note that the trivial labeling maximizes labelweight; the reverse direction of Theorem 18 showed that maximizing labelweight given a fixed download rate implies a linear HSS scheme where the greatest values of d, t can be considered.

This section constructs a family of linear HSS schemes from Hermitian codes; notably, such schemes achieve asymptotically optimal download rate while requiring an amortization parameter that is only linear in s.

▶ **Theorem 22.** *Let ℓ, t, s, d, m be positive integers and q a prime power satisfying $m \geq d$, $s - dt > 0$, and $s = q^3$. Then there exists an explicit t-private, s-server HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ for any non-trivial $\mathcal{F} \subseteq \mathrm{POLY}_{d,m}(\mathbb{F}_{q^2})$ with*

$$\mathit{DownloadRate}(\pi) = 1 - \frac{dt}{s} - \frac{s^{1/3} + 1}{2s^{2/3}}$$

and amortization parameter

$$\ell = s - dt - \frac{s^{2/3} - s^{1/3}}{2}.$$

We note that the above download rate is off of the optimal $1 - dt/s$ by only a $O(s^{-1/3})$ term; *it converges asymptotically to the optimal rate* $1 - dt/s$. Furthermore, it achieves this near-optimal download rate while requiring amortization only linear in s. We place these parameters in the context of [23], [6] in Figure 2.

Table 2 Comparison of Theorem 22 to [23], [6]. When $q = O(1)$ and $s = \omega(1)$, the download rate in Theorem 22 approaches the optimal rate; while the amortization is asmyptotically better.

	[23],[6]	Theorem 22
Download Rate	$1 - dt/s$	$1 - dt/s - O(s^{-1/3})$
Amortization	$(s - dt)\log_{q^2}(s)$	$s - dt - O(s^{2/3})$

We can compare these download rates (and the amortization parameters required to achieve them) for up to 1,000 servers $s \in (dt, 1000]$ in Table 3; we visualize this data in Figure 2.

The key takeaway from these numerical illustrations of Theorem 22 is that *even in non-asymptotic parameter regimes, small concessions in rate result in notable savings in amortization.*

Table 3 Comparison of download rates, amortization parameters from [23], [6] and Theorem 22 when $d = t = 2$.

# Servers	[23], [6]		Theorem 22		% Difference	
	DL Rate	Amort.	DL Rate	Amort.	DL Rate	Amort.
50	0.92	69	0.75	42	-18%	-39%
100	0.96	145	0.83	88	-13%	-39%
200	0.98	294	0.88	182	-10%	-38%
300	0.98	444	0.90	277	-8%	-38%
400	0.99	594	0.91	373	-7%	-37%
500	0.99	744	0.92	469	-7%	-37%
1000	0.99	1494	0.94	951	-5%	-36%

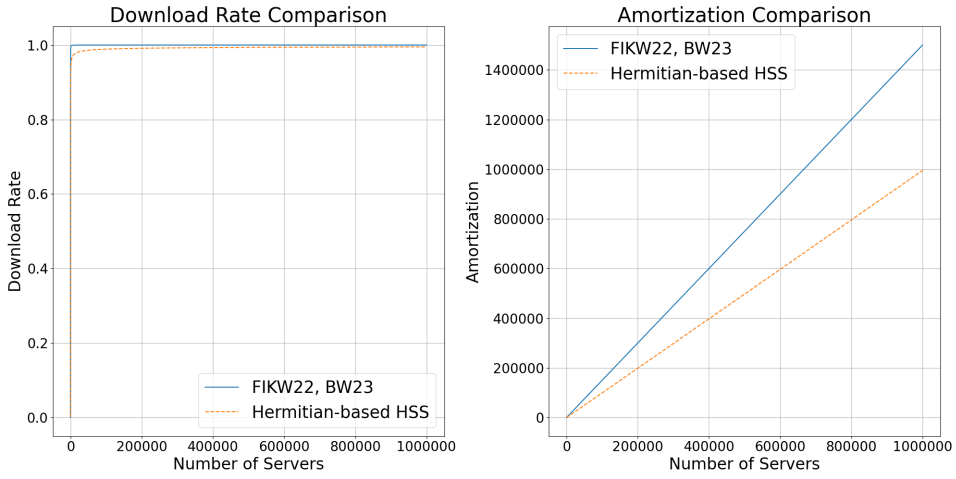

Figure 1 The left (right) plot compares the download rates (amortization parameters) of [23], [6] with that achieved by Theorem 22 when $d = t = 2$. The x-axis denotes the number of servers and ranges from 1 to 1,000,000 to illustrate the asymptotic convergence of Theorem 22 to the optimal rate of [23], [6] at a constant factor less amortization.

4.1 Hermitian Code Definition, Parameters

The construction proceeds by building an optimal labelweight code from Hermitian codes before applying the construction of Theorem 20 to derive the specification of a linear HSS scheme. We begin by recalling the definition and key properties of Hermitian codes. We defer a full treatment of this well-studied family of algebraic geometry codes to [30], [26].

▶ **Definition 23** (Hermitian Curve [30]). *The (affine) Hermitian Curve is given by the planar curve*

$$g(x, y) = y^q + y - x^{q+1}.$$

▶ **Definition 24** (Hermitian Code [26]). *Let $k \in \mathbb{Z}^+$ and let $M \subseteq \mathbb{F}_{q^2}[x, y]$ denote the set of all bi-variate polynomials $f(x, y)$ with total degree $\deg(f) < k$. Denote by*

$$\mathcal{Z} := \left((x, y) \in \mathbb{F}_{q^2}^2 \ : \ g(x, y) = 0\right)$$

the affine rational points of g, and fix any arbitrary ordering of its elements. Then the k-dimensional Hermitian code $\mathcal{H}$ is given by the set of codewords

$$\mathcal{H} := \{\mathrm{ev}_{\mathcal{Z}}(f) \ : \ f \in M\}$$

where $\mathrm{ev}_{\mathcal{Z}}(f) = (f(x, y) : (x, y) \in \mathcal{Z})$ denotes the standard evaluation map.

▶ **Theorem 25** (Hermitian Code Parameters [26]). *The k-dimensional Hermitian code $\mathcal{H}$ is a linear code of length $n = q^3$ and rate k/n with minimum distance*

$$q^3 - k - \frac{q(q-1)}{2} + 1. \tag{1}$$

4.2 Proof of Theorem 22

We first show the following lemma.

▶ **Lemma 26.** *Let $s = q^3, d, t \in \mathbb{Z}^+$ for some prime power q such that $s - dt > 0$. There exists a linear code $\mathcal{C} \subseteq \mathbb{F}_{q^2}^n$ and labeling function $L : [n] \to [s]$ satisfying $\Delta_L(\mathcal{C}) \geq dt + 1$ with rate*

$$R = 1 - \frac{dt}{s} - \frac{s^{1/3} + 1}{2s^{2/3}}$$

and dimension

$$k = s - dt - \frac{s^{2/3} - s^{1/3}}{2}. \tag{2}$$

Proof. Let $\mathcal{H}$ be the k-dimensional Hermitian code defined over alphabet $\mathbb{F}_{q^2}$. By Theorem 25, such a code has length $n = s = q^3$.

Allowing dimension k to be as specified in Equation 2, it follows from Equation 1 that $\Delta(\mathcal{H})$ is given by

$$s - \left(s - dt - \frac{s^{2/3} - s^{1/3}}{2}\right) - \frac{s^{1/3}(s^{1/3} - 1)}{2} + 1 = dt + 1.$$

The rate of $\mathcal{H}$ is given by

$$R_{\mathcal{H}} = \frac{1}{s}\left(s - dt - \frac{s^{2/3} - s^{1/3}}{2}\right) = 1 - \frac{dt}{s} - \frac{s^{1/3} + 1}{2s^{2/3}}.$$

Set $\mathcal{H} = \mathcal{C}$ and $L : [s] \to [s], x \mapsto x$; it immediately follows that $R_{\mathcal{H}} = R_{\mathcal{C}}$ and $\Delta(\mathcal{H}) = \Delta_L(\mathcal{C}) = dt + 1$, as desired. ◀

We are now prepared to prove Theorem 22 by applying Theorem 20.

Proof of Theorem 22. By Lemma 26, there exists a linear code $\mathcal{C} \subseteq \mathbb{F}_{q^2}^s$ and a labeling $L : [s] \to [s], x \mapsto x$ such that $\Delta_L(\mathcal{C}) \geq dt + 1$; furthermore $\mathcal{C}$ has dimension

$$\ell = s - dt - \frac{s^{2/3} - s^{1/3}}{2}$$

and rate

$$R = 1 - \frac{dt}{s} - \frac{s^{1/3}+1}{2s^{2/3}}.$$

By Theorem 20, the existence of such a labelweight code is equivalent to the existence of a linear HSS scheme with corresponding parameters; in particular, there exists a t-private, s-server, $\mathbb{F}_{q^2}$-linear HSS scheme $\pi = \pi(\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ that achieves download rate $\mathsf{DownloadRate}(\pi) = R$ and amortization parameter ℓ. ◀

5 Linear HSS from Goppa Codes

In this section we construct a family of linear HSS schemes from Goppa codes; unlike Theorem 22, these schemes do not achieve asymptotically optimal rate. However, this family of schemes stands apart from those of Theorem 22 by allowing us to compute over the binary field regardless of the number of servers employed. Furthermore, such schemes achieve a super-constant factor of amortization savings at practical server counts. We first state the result before considering its performance in realistic parameter regimes.

▶ **Theorem 27.** *Let ℓ, t, s, d, m, u be positive integers satisfying $m \geq d$, $s - dt > 0$, and $s = 2^u$, where*

$$u > \log_2\left(2(dt)^2 - 4dt + 2(dt+1)\sqrt{(dt)^2 - 2dt + 2} + 3\right).$$

Then there exists an explicit t-private, s-server HSS $\pi = (\mathsf{Share}, \mathsf{Eval}, \mathsf{Rec})$ for some non-trivial $\mathcal{F} \subseteq \mathrm{POLY}_{d,m}(\mathbb{F}_2)$ with

$$\mathit{DownloadRate}(\pi) = 1 - u\frac{dt}{s}$$

and amortization parameter $\ell = s - udt$.

Noting that $u \geq 3$ for all $d, t \in \mathbb{Z}^+$, we see that the download rate does not converge asymptotically to $1 - dt/s$ as the construction of Theorem 22 does; however, we show in Table 4 that for small parameter values, Theorem 27 vastly outperforms Theorem 22 in terms of preserving rate and saving on amortization. In particular, compared to Theorem 27, *the construction of Theorem 27 concedes less rate while delivering an order-of-magnitude savings in amortization* in practical parameter regimes. We illustrate these results graphically in Figure 2.

Table 4 Comparison of Download Rates and Amortization Values with Percentage Differences between FIKW and Goppa.

# Servers	[23],[6]		Theorem 27		% Reduction	
	DL Rate	Amortize	DL Rate	Amortize	DL Rate	Amortize
64	0.93	360	0.65	42	-31%	-88%
128	0.96	868	0.82	106	-15%	-88%
256	0.98	2016	0.91	234	-7%	-88%
512	0.99	4572	0.96	490	-3%	-89%
1024	0.99	10200	0.98	1002	-1.8%	-90%
2048	0.99	22484	0.99	2026	-0.9%	-91%

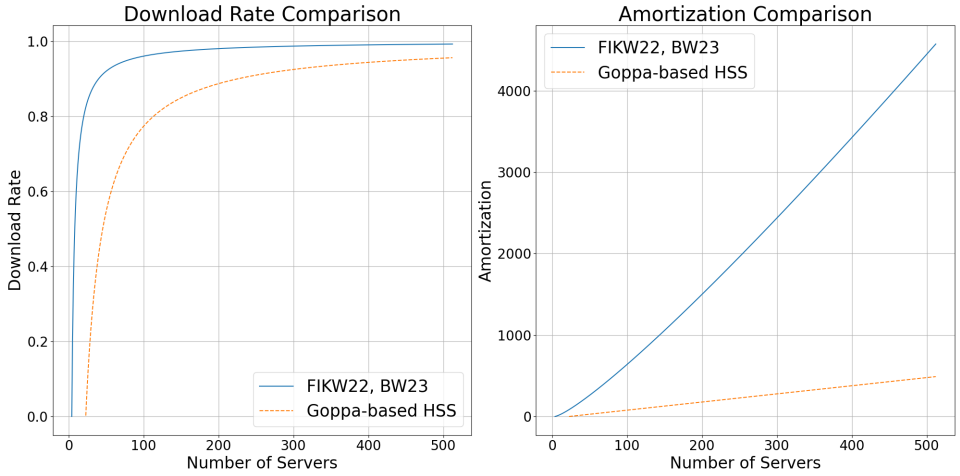

Figure 2 The left (right) plot compares the download rates (amortization parameters) of [23], [6] with that achieved by Theorem 27 when $d = t = 2$. The x-axis represents the number of servers and ranges from 1 to 512. This emphasizes the super-constant amortization savings of Theorem 27 at practical parameter regimes relative to [23], [6], with small concessions to rate.

5.1 Goppa Code Definition, Parameters

The proof of Theorem 27 is constructive; it proceeds by building an optimal labelweight code from Goppa codes before applying the construction of Theorem 20 to arrive at a linear HSS scheme with the desired properties. We begin by recalling the definition and key properties of binary Goppa codes, deferring a fuller treatment to [5], [24].

▶ **Definition 28** (Goppa Polynomial [5]). *For some $n \in \mathbb{Z}^+$, fix $V = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_{2^u}$, $u \in \mathbb{Z}^+$. A Goppa polynomial is a polynomial*

$$g(x) = g_V(x) = g_0 + g_1 x + \cdots + g_r x^r \in \mathbb{F}_{2^u}[x]$$

satisfying $\deg(g) = r$ *and* $g(\alpha_i) \neq 0$ *for all* $\alpha_i \in V$.

Given the definition of the Goppa polynomial above, we can define a binary Goppa code.

▶ **Definition 29** (Goppa Codes [5]). *Let $n, u, r \in \mathbb{Z}^+$. Fix $V = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_{2^u}$, $u \in \mathbb{Z}^+$ and let $g_V \in \mathbb{F}_{2^u}[x]$ be a Goppa polynomial of degree r. Then the Goppa code is the set of codewords given by*

$$\Gamma_{n,u,r} = \Gamma_{n,u,r}(g, V) := \left\{ \mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_2^n \ : \ \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \mod g(x) \right\}$$

.

The parameters of Goppa codes are given by the following theorem.

▶ **Theorem 30** (Goppa Code Parameters [5]). *For $n, u, r \in \mathbb{Z}^+$ let $\Gamma = \Gamma_{n,u,r}$ be a binary Goppa code as in Definition 29. Then Γ is a linear code of length n, dimension $k \geq n - ur$, and minimum distance $d(\Gamma) \geq r + 1$.*

The parameters given by Theorem 30 only allow us to determine rate and minimum distance up to a lower bound, making it difficult to ascertain download rate and amortization when used to construct linear HSS schemes. Fortunately, these lower bounds are known to be sharp under additional assumptions. The following theorem gives one such instance.

▶ **Theorem 31** ([31]). *Fix $u, r \in \mathbb{Z}^+$ satisfying*

$$2r - 2 < \frac{2^u - 1}{2^{u/2}} \tag{3}$$

and let $g \in \mathbb{F}_{2^u}[x]$ be a Goppa polynomial of degree r with no repeated roots. Set $V = \mathbb{F}_{2^u}$ and let $\Gamma = \Gamma_{2^u,u,r}(g, V)$ be a Goppa code as in Definition 29. Then Γ is a binary linear code with dimension precisely $k = n - ur$.

We observe that, performing the appropriate manipulations, Equation 3 is satisfied for all

$$u \geq \max\Big\{ \Big\lceil \log_2\Big(2r^2 - 4r + 2(r+1)\sqrt{r^2 - 2r + 2} + 3\Big)\Big\rceil, \log_2\Big(2r^2 - 4r + 2(r+1)\sqrt{r^2 - 2r + 2} + 3\Big) + 1\Big\}. \tag{4}$$

5.2 Proof of Theorem 27

In this section we prove Theorem 27. We first show the following lemma.

▶ **Lemma 32.** *Let $s, d, t, u \in \mathbb{Z}^+$ satisfy $s - dt > 0$ and $s = 2^u$, where*

$$u = \max\Big\{ \Big\lceil \log_2\Big(2(dt)^2 - 4dt + 2(dt+1)\sqrt{(dt)^2 - 2dt + 2} + 3\Big)\Big\rceil, \log_2\Big(2(dt)^2 - 4dt + 2(dt+1)\sqrt{(dt)^2 - 2dt + 2} + 3\Big) + 1\Big\}. \tag{5}$$

There exists a linear code $\mathcal{C} \subseteq \mathbb{F}_2^n$ and labeling function $L : [n] \to [s]$ satisfying $\Delta_L(\mathcal{C}) \geq dt + 1$ with rate $R = 1 - udt/s$ and dimension $\ell = s - udt$.

Proof. Fix $V = \mathbb{F}_{2^u}$ and let $g \in \mathbb{F}_{2^u}[x]$ be an irreducible polynomial of degree $r = dt$; set $n = 2^u$. Let $\Gamma = \Gamma_{n,u,r}(g, V)$ be the binary Goppa code given by Definition 29. It follows from Equation 5 and the observation of Equation 4 that Γ has dimension $k = n - ur = s - udt$. It follows from Theorem 30 that Γ has minimum distance $d(\Gamma) \geq r + 1 = dt + 1$. Set $\mathcal{C} = \Gamma$ and define $L : [n] \to [s], x \mapsto x$ to be the identity labeling. It immediately follows that $\mathcal{C}$ has the desired rate and dimension. ◀

It is now straightforward to prove Theorem 27 by leveraging Theorem 20.

Proof of Theorem 27. By Lemma 32, there exists a linear code $\mathcal{C} \subseteq \mathbb{F}_2^s$ and a labeling $L : [s] \to [s]$ such that $\Delta_L(\mathcal{C}) \geq dt + 1$; furthermore $\mathcal{C}$ has dimension $\ell = s - udt$ and rate $R = 1 - udt/s$. By Theorem 20, the existence of such a labelweight code is equivalent to the existence of a linear HSS scheme with corresponding parameters. ◀

References

1 Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *STACS 90*, pages 37–48, 1990.

2 Donald Beaver, Joan Feigenbaum, Joe Kilian, and Phillip Rogaway. Security with low communication overhead. In *CRYPTO '90*, pages 62–76, 1990.

3 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, 1988.

4 Josh Cohen Benaloh. Secret sharing homomorphisms: Keeping shares of A secret sharing. In Andrew M. Odlyzko, editor, *CRYPTO '86*, pages 251–260, 1986.

5 Elwyn Berlekamp. Goppa codes. *IEEE Transactions on Information Theory*, 19(5):590–592, 1973.

6 Keller Blackwell and Mary Wootters. A characterization of optimal-rate linear homomorphic secret sharing schemes, and applications. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

7 Keller Blackwell and Mary Wootters. Improved trade-offs between amortization and download bandwidth for linear hss. *arXiv preprint arXiv:2403.08719*, 2024.

8 Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In *CRYPTO*, pages 489–518, 2019.

9 Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, and Michele Orrù. Homomorphic secret sharing: optimizations and applications. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2105–2122, 2017.

10 Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *EUROCRYPT 2015, Part II*, pages 337–367, 2015.

11 Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 509–539. Springer, 2016. doi:10.1007/978-3-662-53018-4_19.

12 Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1292–1303. ACM, 2016. doi:10.1145/2976749.2978429.

13 Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of homomorphic secret sharing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 21:1–21:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.21.

14 Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In *EUROCRYPT 2019, Part II*, pages 3–33, 2019.

15 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, 1988.

16 Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *J. ACM*, 1998.

17 Geoffroy Couteau and Pierre Meyer. Breaking the circuit size barrier for secure computation under quasi-polynomial LPN. In *EUROCRYPT 2021, Part II*, pages 842–870, 2021.

18 Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In Joe Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 342–362. Springer, 2005. doi:10.1007/978-3-540-30576-7_19.

19 Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT*, 2000.

20 Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear mpc from sparse lpn. In *Annual International Cryptology Conference*, pages 315–348. Springer, 2023.

21 Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 93–122. Springer, 2016. doi:10.1007/978-3-662-53015-3_4.

22 Nelly Fazio, Rosario Gennaro, Tahereh Jafarikhah, and William E. Skeith III. Homomorphic secret sharing from Paillier encryption. In *Provable Security*, 2017.

23 Ingerid Fosli, Yuval Ishai, Victor I Kolobov, and Mary Wootters. On the download rate of homomorphic secret sharing. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

24 Valerii Denisovich Goppa. Codes associated with divisors. *Problemy Peredachi Informatsii*, 13(1):33–39, 1977.

25 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential coding theory*. Draft from http://www.cse.buffalo.edu/atri/courses/coding-theory/book, 2019.

26 James William Peter Hirschfeld, Gábor Korchmáros, and Fernando Torres. *Algebraic curves over a finite field*, volume 20. Princeton University Press, 2008.

27 Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64, 1989.

28 Claudio Orlandi, Peter Scholl, and Sophia Yakoubov. The rise of paillier: Homomorphic secret sharing and public-key silent OT. In *EUROCRYPT 2021, Part I*, pages 678–708, 2021.

29 Lawrence Roy and Jaspal Singh. Large message homomorphic secret sharing from DCR and applications. In *CRYPTO 2021, Part III*, pages 687–717, 2021.

30 Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Publishing Company, Incorporated, 2nd edition, 2008.

31 M. Van der Vlugt. The true dimension of certain binary goppa codes. *IEEE Transactions on Information Theory*, 36(2):397–398, 1990. doi:10.1109/18.52487.

A Linear HSS from Random Codes

In this section we show that the natural random coding approach does not appear to yield linear HSS schemes that meaningfully outperform the $\ell = O(s \log(s))$ amortization parameter required by [23], [6]. Indeed, the standard argument established that random linear codes correspond to linear HSS schemes that can attain good download rates, but – like [6, 23] – only with large amortization parameters.

A.1 Notation

To justify the notion that "random labelweight codes don't outperform Reed-Solomon codes in linear HSS amortization", we proceed by generalizing the well-known Gilbert-Varshamov Bound to the labelweight setting. One standard proof of this result (see, e.g., [25]) analyzes the distance of a random linear code, and we follow the same path here. We first introduce some notation.

▶ **Definition 33** (Labelweight Ball). *Let $L : [n] \to [s]$ be a surjective labeling. We define the labelweight ball $B_L(r)$ of radius $0 \leq r \leq n$ to be the set*

$$B_L(r) := \left\{ c \in \mathbb{F}_q^n \ : \ \Delta_L(c) \leq r \right\}$$

and define the volume of the labelweight ball to be $\mathrm{Vol}_L(r) = |B_L(r)|$.

For the purposes of our analysis, we consider only a fixed labeling function.

▶ **Assumption 1.** *Let $n, s, w \in \mathbb{Z}^+$ such that $n = sw$. In this section we will only consider the labeling*

$$L : [n] \to [s], \ x \mapsto \left\lceil \frac{x}{s} \right\rceil .$$

When paired with a code of length n, this balanced labeling simply labels the first w coordinates with 1, the second w coordinates with 2, and continues analogously until the last w coordinates are labeled with s. Intuitively, for fixed $n, s \in \mathbb{Z}^+$, such a balanced labeling pattern maximizes labelweight in expectation over random linear codes.

Under this fixed, balanced labeling function of Assumption 1, we have the following algebraic formulation of labelweight ball volume.

▶ **Observation 34.** *Let $n, s, w, r \in \mathbb{Z}^+$ such that $n = sw$ and $0 \leq r \leq n$. Then*

$$\mathrm{Vol}_L(r) = |B_L(r)| = \sum_{i=0}^{r} \binom{s}{i} (q^w - 1)^i .$$

Finally, we will need to define relative labelweight for labelweight codes, which is the natural analogue of relative minimum distance for linear codes.

▶ **Definition 35** (Relative Labelweight)**.** *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code and $L : [n] \to [s]$ a surjective labeling such that $\Delta_L(\mathcal{C}) = d$. We define the relative labelweight of $\mathcal{C}$ to be $\delta = d/s$.*

A.2 Generalization of q-ary Entropy

In the standard proof of the Gilbert-Varshamov bound, the volume of a Hamming ball is estimated by the q-ary entropy function. To generalize the proof to labelweight, we introduce the following generalization of the q-ary entropy function, which captures the volume of labelweight balls.

▶ **Definition 36** (Generalized q-ary Entropy)**.** *Let $q \geq 2$, $w \geq 1$. For $x \in (0, 1)$, we denote by $H_{q,w}(x)$ the generalized q-ary entropy function:*

$$H_{q,w}(x) = x \log_q(q^w - 1) - x \log_q(x) - (1 - x) \log_q(1 - x),$$

where $H_{q,w}(0), H_{q,w}(1)$ are defined as the limit of $H_{q,w}$ as $x \to 0, 1$, respectively.

Note that the case where $w = 1$ is the standard q-ary entropy function. We notice that, when properly normalized, the generalized entropy function can be approximated linearly.

▶ **Observation 37.** *For all $x \in [0, 1 - 1/q^w]$, $x \leq w^{-1} H_{q,w}(x) \leq x + \log_q(2^{1/w})$.*

Proof. Observe that

$$\begin{aligned} g(x) := w^{-1} H_{q,w}(x) - x &= w^{-1} \left(x \log_q(q^w - 1) - x \log_q(x) - (1 - x) \log_q(1 - x) \right) - x \\ &\leq w^{-1} \left(-x \log_q(x) - (1 - x) \log_q(1 - x) \right). \end{aligned}$$

Since $-x \log_q(x) - (1 - x) \log_q(1 - x)$ is a concave function which attains its maximal value when $x = 1/2$, it follows that $w^{-1} H_{q,w}(x) - x \leq w^{-1} \log_q(2)$ as desired. The lower bound follows from observing that g is itself a concave function, since

$$g''(x) = -\frac{1}{w \cdot x \cdot (1 - x) \cdot \ln(q)} \leq 0 \; \forall x \in [0, 1 - 1/q^w]$$

and that its values at the endpoints of the domain $[0, 1 - 1/q^w]$ are non-negative; i.e., $g(0), g(1 - 1/q^w) \geq 0$. ◀

Equipped with this definition, our goal becomes to express the volume of a given labelweight ball in terms of the generalized entropy function. To do so, we note two helpful relations; we omit the proofs, which are elementary algebraic manipulations.

▶ **Observation 38.** *Let $s, p \in \mathbb{R}$ such that $s, p \geq 0$. Then*

$$q^{-sH_{q,w}(p)} = (1-p)^{(1-p)s} \left(\frac{p}{q^w - 1}\right)^{ps}$$

▶ **Observation 39.** *Let $w \in \mathbb{Z}^+$ and $p \in [0, 1)$ satisfy $0 \leq p \leq 1 - 1/q^w$. Then*

$$\frac{p}{(1-p)(q^w - 1)} \leq 1.$$

We now give the volume of a labelweight ball in terms of the generalized entropy function.

▶ **Lemma 40.** *Let $s, w \in \mathbb{Z}^+$ and $p \in [0, 1)$ satisfy $0 \leq p \leq 1 - 1/q^w$ and $ps \in \mathbb{Z}^+$. Then*

$$\mathrm{Vol}_L(ps) \leq q^{sH_{q,w}(p)}.$$

Proof. Observe that

$$1 = (p + (1-p))^s = \sum_{i=0}^{s} \binom{s}{i} p^i (1-p)^{s-i} \geq \sum_{i=0}^{ps} \binom{s}{i} p^i (1-p)^{s-i}.$$

Multiplying through by $1 = (q^w - 1)^i / (q^w - 1)^i$ and applying Observation 39 yields

$$1 \geq \sum_{i=0}^{ps} \binom{s}{i} (q^w - 1)^i (1-p)^s \left(\frac{p}{(1-p)(q^w - 1)}\right)^{ps}.$$

Finally, applying Observation 38 yields

$$\begin{aligned} 1 &\geq \sum_{i=0}^{ps} \binom{s}{i} (q^w - 1)^i q^{-sH_{q,w}(p)} \\ &= \mathrm{Vol}_L(ps) q^{-sH_{q,w}(p)}. \end{aligned}$$

◀

A.3 Gilbert-Varshamov Bound for Random Labelweight Codes

We are finally equipped to prove a generalization of the Gilbert-Varshamov bound for labelweight codes. This generalization will quantify the rate, and labelweight trade-off we can guarantee through random linear codes; viewed through the lens of Theorem 20, this tells us the download rate and amortization parameters that can be guaranteed by linear HSS scheme constructed from random linear codes.

▶ **Theorem 41.** *For $q \geq 2$, let $n, s, w \in \mathbb{Z}^+$ satisfy $n = sw$. Let $\delta \in [0, 1 - 1/q^w]$ satisfy $\delta s \in \mathbb{Z}^+$. For $\varepsilon \in [0, 1 - H_{q,w}(\delta)]$, let*

$$k = n - sH_{q,w}(\delta) - n\varepsilon \tag{6}$$

and let $G \in \mathbb{F}_q^{k \times n}$ be chosen uniformly at random.

Then with probability $> 1 - q^{-\varepsilon n}$, G is the generator matrix of a length n, dimension k, and relative labelweight $\geq \delta$ linear code with rate

$$R = 1 - \frac{sH_{q,w}(\delta)}{n} - \varepsilon.$$

Note that when $n = s$ and $w = 1$, Theorem 41 becomes the standard Gilbert-Varshamov Bound. Before we show the proof of Theorem 41, we interpret its statement in terms of linear HSS parameters.

▶ **Example 42.** Let $s, d, t \in \mathbb{Z}^+$ satisfying $s - dt > 0$ parameterize a linear HSS scheme as in Definition 7. Let s be as stated in Theorem 41 and set $\delta = (dt+1)/s$.

For the sake of illustration, suppose $w = \log_q(s)$ and $\varepsilon > 0$ a negligible constant. Let $\mathcal{C}$ denote the linear code with properties guaranteed by Theorem 41 and let π denote the t-private, s-server linear HSS constructed from $\mathcal{C}$ as in Theorem 20. Applying Observation 37 to Theorem 20, π has download rate at most

$$\mathsf{DownloadRate}(\pi) \leq 1 - \frac{dt+1}{s} - \varepsilon = 1 - \frac{dt}{s} - O(s^{-1})$$

with amortization parameter *at least*

$$\ell \geq (1-\varepsilon)s\log_q(s) - s\log_q(2) - (dt+1)\log_q(s) = \Omega(s\log(s))$$

for sufficiently small ε. In particular, we note that such a construction has an amortization parameter (at least) on the same $\Omega(s\log(s))$ order as that of [23], [6], while achieving a rate comparable to that of our Hermitian code-based construction of Theorem 22. We summarize this situation in Table 5.

Table 5 Comparison of Theorem 22 to Example 42.

	Thm. 22 (Hermitian code-based)	Ex. 42 (Random code-based)
Download Rate	$1 - dt/s - O(s^{-1/3})$	$\leq 1 - dt/s - O(s^{-1})$
Amortization	$s - dt - O(s^{2/3})$	$\Omega(s\log(s))$

We conclude this section by proving Theorem 41.

Proof of Theorem 41. Let $\mathcal{C} = \{\mathbf{m}G \ : \ \mathbf{m} \in \mathbb{F}_q^k\}$ be the linear code generated by G. It suffices to show that $\Delta_L(\mathbf{m}G) \geq d$ for all non-zero $\mathbf{m}$.

Accordingly, let $\mathbf{m} \in \mathbb{F}_q^k$ be a uniformly random non-zero vector; then $\mathbf{m}G$ is uniformly distributed over $\mathbb{F}_q^n$. It follows from Lemma 40 that

$$\Pr\left[\delta_L(\mathbf{m}G) < d\right] = \frac{\mathrm{Vol}_L(d-1)}{q^n} \leq \frac{q^{sH_{q,w}(\delta)}}{q^n} = q^{-k}\, q^{-n\varepsilon}.$$

Taking the Union Bound over all $\mathbf{m} \in \mathbb{F}_q^k$ yields the observation that with probability $1 - q^{-n\varepsilon}$, $\Delta_L(\mathcal{C}) \geq d$ as desired. ◀