Weighted Secret Sharing from Wiretap Channels

Fabrice Benhamouda
Algorand Foundation, New York, NY, USA

Shai Halevi
Algorand Foundation, New York, NY, USA

Lev Stambler
Independent Researcher, NJ, USA

Abstract

Secret-sharing allows splitting a piece of secret information among a group of shareholders, so that it takes a large enough subset of them to recover it. In *weighted* secret-sharing, each shareholder has an integer weight, and it takes a subset of large-enough weight to recover the secret. Schemes in the literature for weighted threshold secret sharing either have share sizes that grow linearly with the total weight, or ones that depend on huge public information (essentially a garbled circuit) of size (quasi)polynomial in the number of parties.

To do better, we investigate a relaxation, (α, β)-ramp weighted secret sharing, where subsets of weight βW can recover the secret (with W the total weight), but subsets of weight αW or less cannot learn anything about it. These can be constructed from standard secret-sharing schemes, but known constructions require long shares even for short secrets, achieving share sizes of $\max\left(W, \frac{|\text{secret}|}{\epsilon}\right)$, where $\epsilon = \beta - \alpha$. In this note we first observe that simple rounding let us replace the total weight W by N/ϵ, where N is the number of parties. Combined with known constructions, this yields share sizes of $O\big(\max(N, |\text{secret}|)/\epsilon\big)$.

Our main contribution is a novel connection between weighted secret sharing and wiretap channels, that improves or even eliminates the dependence on N, at a price of increased dependence on $1/\epsilon$. We observe that for certain additive-noise $(\mathcal{R}, \mathcal{A})$ wiretap channels, any semantically secure scheme can be naturally transformed into an (α, β)-ramp weighted secret-sharing, where α, β are essentially the respective capacities of the channels $\mathcal{A}, \mathcal{R}$. We present two instantiations of this type of construction, one using Binary Symmetric wiretap Channels, and the other using additive Gaussian Wiretap Channels. Depending on the parameters of the underlying wiretap channels, this gives rise to (α, β)-ramp schemes with share sizes $|\text{secret}| \cdot \log N/\text{poly}(\epsilon)$ or even just $|\text{secret}|/\text{poly}(\epsilon)$.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Security and privacy → Information-theoretic techniques

Keywords and phrases Secret sharing, ramp weighted secret sharing, wiretap channel

Digital Object Identifier 10.4230/LIPIcs.ITC.2023.8

Related Version *Full Version*: IACR ePrint 2022/1578 [8]

Funding *Lev Stambler*: Work done while in Algorand Foundation, USA.

Acknowledgements We thank the anonymous reviewers for their comments, which vastly improved this work.

1 Introduction

Secret sharing [24, 10] allows a dealer to split some secret information among multiple parties, giving each party an individual share, so that large enough subsets of shareholder can recover the secret, but small subsets cannot learn any partial information about it. Such schemes are typically parametrized by the number of parties N and a threshold $T \leq N$, such that it takes at least T parties to recover the secret.

4th Conference on Information-Theoretic Cryptography (ITC 2023).
Editor: Kai-Min Chung; Article No. 8; pp. 8:1–8:19
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Weighted secret sharing (WSS) is similar, except that each shareholder j has an integer weight w_j, it takes a "heavy enough" subsets to recover the secret, while "light" subsets cannot learn any partial information about it. The threshold $T \in [N]$ is replaced by $\tau \in (0, 1)$, such that it takes shareholders of aggregate weight τW to recover the secret (where W is the total weight, $W = \sum_{j \in [N]} w_j$).

One method of implementing WSS is to rely on standard secret-sharing with $N' = W$ and $T' = \tau W$, giving w_j shares to a shareholder j with weight w_j. While this solution can achieve good rate for long secrets (see Section 3.1), it is very wasteful for short ones, as its share sizes grow linearly with the weight. Prior work on weighted secret sharing explored other solutions (e.g., using Chinese remaindering) or limited models (e.g., specific weight hierarchies). But they all still feature either linear dependency of the share-size on W, severe restrictions to the access structures that can be realized, or huge public information that must be broadcasted to everyone alongside the individual shares. (See more discussion in Section 1.2 below.)

In an attempt to do better, in this work we consider the relaxed model of ramp secret-sharing [9], that has a fuzzy threshold. Specifically, an (α, β)-ramp weighted secret sharing scheme allows any subset of aggregate weight at least βW to recover the secret, but subsets of weight αW or less cannot learn any information about it. Such gaps were considered often in the literature for standard secret-sharing schemes, but to our knowledge were not studied in the context of weighted secret sharing.

It is not hard to see (and we describe it explicitly in Section 3) that this relaxation enables shorter secrets, just by keeping only a $1/\epsilon$ precision for the weights, where $\epsilon = \beta - \alpha$. Rather than linear dependence on the weights, we now get linear dependence on N/ϵ (where the dependence on the number of parties N is due to the accumulation of rounding errors in this limited-precision approximation).

Beyond this simple observation, the main technical meat in this work is a novel blueprint for (α, β)-ramp WSS schemes, by exploring a surprising connection to secure transmission schemes for wiretap channels. These constructions reduce or even eliminate the dependence on N, at the price of potentially worse (but still polynomial) dependence on $1/\epsilon$. We note that the field of wiretap coding is an ongoing line of research with an aim of decreasing dependence on $1/\epsilon$. Any advances in wiretap coding can easily be applied to WSS with our construction.

1.1 Overview of Our Techniques

The starting point for our new blueprint is the following approach: On input $\boldsymbol{s}$, the dealer gives each shareholder j an independent noise vector $\boldsymbol{e}_j$, whose magnitude depends on their weight, and publishes the value $\boldsymbol{g} = \mathsf{Enc}(\boldsymbol{s}) + \sum_j \boldsymbol{e}_j$, where $\mathsf{Enc}(\cdot)$ is some encoding function.[1] Given the public $\boldsymbol{g}$ and their individual $\boldsymbol{e}_j$'s, the only information that a set T of shareholder has on the secret $\boldsymbol{s}$ is the value

$$\boldsymbol{g}_T = \boldsymbol{g} - \sum_{j \in T} \boldsymbol{e}_j = \mathsf{Enc}(\boldsymbol{s}) + \sum_{j \notin T} \boldsymbol{e}_j.$$

We can therefore associate with each subset T an additive-noise channel $C_T : x \mapsto x + \sum_{j \notin T} \boldsymbol{e}_j$, such that the information that T learns about $\boldsymbol{s}$ is exactly the received value $C_T(\mathsf{Enc}(\boldsymbol{s}))$. We are seeking an encoding function $\mathsf{Enc}(\cdot)$ so that:

[1] Publishing $\boldsymbol{g}$ can be done by sending it to all the shareholders, which will only double the size of the shares that each one holds. It may be possible to use information dispersal to do even better.

- Any qualified set S can recover $\boldsymbol{s}$ from $C_S(\mathsf{Enc}(\boldsymbol{s}))$;
- For any unqualified set T, seeing $C_T(\mathsf{Enc}(\boldsymbol{s}))$ yields no information on $\boldsymbol{s}$.

Intuitively, the smaller (or "lighter") the set is, the more error components it is missing, so the more noisy its channel will be. Consider now $\mathcal{R}$ which is "the most noisy channel" for any qualified set, and $\mathcal{A}$ which is "the least noisy channel" for any unqualified set.

We can hope that $\mathcal{R}$ is less noisy than $\mathcal{A}$, and use a good transmission scheme for the wiretap channel $(\mathcal{R}, \mathcal{A})$, with receiver channel $\mathcal{R}$ and adversary channel $\mathcal{A}$. (Recall that a wiretap scheme for a pair of channels $(\mathcal{R}, \mathcal{A})$ consists of an encoding function $\mathsf{Enc}(\cdot)$ such that a secret $\boldsymbol{s}$ can be recovered from $\mathcal{R}(\mathsf{Enc}(\boldsymbol{s}))$ whp, but where $\mathcal{A}(\mathsf{Enc}(\boldsymbol{s}))$ yields almost no information on $\boldsymbol{s}$.)

Trying to flesh out this approach, we need to associate an error distribution $\mathcal{D}_{w_j}$ to every weight $w_j \in \mathbb{N}$, so that whenever $\sum_{j\in A} w_j > \sum_{j\in B} w_j$ it holds that $\sum_{j\in A} \mathcal{D}_{w_j}$ has "more error" than $\sum_{j\in B} \mathcal{D}_{w_j}$. Then we need to find two concrete channels $\mathcal{R}, \mathcal{A}$ such that

- $\mathcal{R}$ is at least as noisy as C_Q for any qualified set Q with weight $\geq \beta W$.
- $\mathcal{A}$ is at most as noisy as C_U for any unqualified set U with weight $\leq \alpha W$.

If $\mathcal{R}$ is less noisy than $\mathcal{A}$, then we can use a good transmission scheme for the wiretap channel $(\mathcal{R}, \mathcal{A})$ to implement our (α, β)-ramp WSS scheme. The parameters of this WSS scheme can be derived from those of the underlying wiretap scheme.

1.1.1 Binary Symmetric Channels

Trying to instantiate this approach with binary symmetric channels, we associate with each weight w_j an error probability p_j and the corresponding Bernoulli random variable

$$\mathcal{D}_j = \begin{cases} 1 & \text{with probability } p_j \\ 0 & \text{with probability } 1 - p_j. \end{cases}$$

One problem to overcome is that for "the most natural mapping" of weights to probabilities, the error probability does not add up linearly: If we set (say) $p_j = w_j/W$, it is not hard to find instances where $\sum_{j\in A} w_j > \sum_{j\in B} w_j$ and yet $\sum_{i\in A} \mathcal{D}_j \bmod 2$ has smaller error probability than $\sum_{j\in B} \mathcal{D}_j \bmod 2$, as the following example shows.

A problematic example

Consider three parties with $w_1 = w_2 = 13$ and $w_3 = 24$, so $W = 50$ and we have $\Pr[\mathcal{D}_1 = 1] = \Pr[\mathcal{D}_2 = 1] = 13/50 = 0.26$ and $\Pr[\mathcal{D}_3 = 1] = 24/50 = 0.48$. Let $A = \{1, 2\}$ and $B = \{3\}$, so the aggregate weight of A is 26, larger that the weight of B which is 24. On the other hand, we have

$$\Pr[\mathcal{D}_1 \oplus \mathcal{D}_2 = 1] = 0.26 + 0.26 - 0.26^2 = 0.4525 < 0.48 = \Pr[\mathcal{D}_3 = 1],$$

so the error rate for A is *lower* that for B.

Clearly, the reason for this example is the cancellation due to the term 0.26^2, namely the fact that the error probabilities do not simply add up. This cancellation effect can be reduced it by scaling down the probabilities, setting $p_j = \gamma w_j/W$ for some $\gamma < 1$ (that may depend on α, β). For example, if we set $p_j = w_j/2W$ rather than $p_j = w_j/W$, then we get $\Pr[\mathcal{D}_1 = 1] = \Pr[\mathcal{D}_2 = 1] = 0.13$ and $\Pr[\mathcal{D}_3 = 1] = 0.24$, and therefore

$$Pr[\mathcal{D}_1 \oplus \mathcal{D}_2 = 1] = 0.13 + 0.13 - 0.13^2 = 0.2431 > 0.24 = \Pr[\mathcal{D}_3 = 1].$$

While this can be made to work, it has a drawback that the error rates of the scaled $\mathcal{R}$ and $\mathcal{A}$ become quite close, of distance only $O(\epsilon^2)$ (where $\epsilon = \beta - \alpha$). This would require the codes of fairly large block-length, making the share-size a large polynomial in $1/\epsilon$. Instead, we describe here a different variant that was pointed out to us by the anonymous reviewers, that improves the dependence on $1/\epsilon$ by using a better mapping from weights to probabilities.

1.1.1.1 The BSC construction

The BSC-based construction that we present in Section 5 gives a weight-w_j shareholder an error variable $\mathcal{D}(w_j)$ which is the sum modulo 2 of w_j IID Bernoulli random variables, all with the same head probability of $\tau < 1/2$ (where τ can depend on α, and β and the total weight $W = \sum_j w_j$). With this definition, it is clear that we get additivity, namely $\mathcal{D}(w_1 + w_2) = \mathcal{D}(w_1) + \mathcal{D}(w_2) \pmod 2$. Therefore, the error sum of a shareholder set with cumulative weight w is exactly $\mathcal{D}(w)$, Also, it is not hard to show that for this construction, the head probability of $\mathcal{D}(w_j)$ is

$$\Pr[\mathcal{D}(w_j) = 1] = \frac{1}{2} \cdot (1 - \exp(-\gamma \cdot w_j/W))\,,$$

where γ is some constant that depends on τ. As we show in Section 5, optimizing the constant γ in this construction yields a wiretap channel $(\mathcal{R}, \mathcal{A})$ where the capacity gap between $\mathcal{R}$ and $\mathcal{A}$ is $\Theta(\beta - \alpha)$.

1.1.2 Additive Gaussian Channels

Another natural attempt to instantiate our blueprint is using additive white Gaussian noise (AWGN) channels. For these channels, the noise is natively additive: adding Gaussian variables with variance σ_1^2 and σ_2^2 yields another Gaussian with variance $\sigma_1^2 + \sigma_2^2$. The AWGN-noise construction therefore associates each weight, $w \in \mathbb{N}$ with the Normal random variable $\mathcal{N}(0, w/W)$, i.e. zero-mean with variance w/W (stdev $= \sqrt{w/W}$). Due to additivity, the aggregate random variable for a set A is itself a Normal variable,

$$\sum_{j \in A} \mathcal{N}(0, w_j/W) = \mathcal{N}(0, \sum_{j \in A} w_j/W).$$

This implies that whenever S has higher weight than T, the channel C_T has more error than the channel C_S.

For any $\beta > \alpha$, we can therefore construct an (α, β)-ramp WSS scheme from a good transmission scheme for the AWGN wiretap channel $(\mathcal{R}, \mathcal{A})$, where

$$\mathcal{R} : x \mapsto x + \mathcal{N}(0, 1 - \beta) \quad \text{and} \quad \mathcal{A} : x \mapsto x + \mathcal{N}(0, 1 - \alpha).$$

Indeed, since $\beta > \alpha$ then $\mathcal{A}$ is more noisy than $\mathcal{R}$.

One problem to solve when using AWGN channels is that they natively deal with real numbers with infinite precision, whereas we can only use finite precision for our construction. In Appendix A we therefore sketch an approach that uses discrete Gaussians instead. That construction achieves somewhat worse rate than the BSC construction for long secrets, but it can plausibly offer concrete parameter benefits for short secrets.

1.2 Prior Work

Ramp secret-sharing (without weights) was introduced by Blakley and Meadows [9]. A textbook construction for a ramp-scheme with good rate based on standard "packed secret sharing" can be found, e.g., in [12, 11.4.2] (and is described in Section 3.1 below).

Some early work on weighted secret sharing was cast against the backdrop of general access structures. Beimel et al. [3] characterized the weighted (strict) thresholds access structures that admit *ideal* schemes, where the share size is equal to the secret size, proving that only few specific threshold structures can be realized this way.

Beimel and Weinreb [4] showed that any threshold access structure can be realized using shares of size $\mathrm{quasiPoly}(N \log W)$ times the secret size, or even just $\mathsf{poly}(N \log W) \cdot \lambda$ if computational security is enough (λ is the security parameter). They did that by describing monotone circuits that compute every threshold function, and using known monotone-circuits-to-secret-sharing compilers [7, 27].[2] Works such as [16] and [25] propose an explicit scheme for hierarchical threshold structures, those are solving a different (albeit somewhat related) problem than ours.

Another notable prior work is due to Zou et al. [29], they use the Chinese Remainder Theorem to improve some efficiency parameters of weighted multi-secret sharing, but secret sizes are still the same as in the simple scheme based on Shamir sharing.

Also, noisy channels were used in many prior works as a tool for achieving secure computation, starting with [14, 13]. The goals in that line of works are quite different from ours, however, and the connection that we draw between ramp secret-sharing and wiretap channels is new.

Organization

We present some background in Section 2, then define (α, β)-ramp WSS and describe a simple rounding-based protocol for realizing it in Section 3. We formulate our blueprint for WSS schemes from wiretap schemes in Section 4, then describe instantiations of this blueprint from binary symmetric channels in Section 5 and from additive white Gaussian noise channels in Appendix A.

2 Background

Notations. For an integer n, we denote $[n] = \{1, 2, \ldots, n\}$. The ℓ'th entry in a vector $\boldsymbol{e}$ is denoted $\boldsymbol{e}[\ell]$. For two distributions $\mathcal{D}, \mathcal{E}$, we denote by $SD(\mathcal{D}, \mathcal{E})$ their statistical distance. Namely $SD(\mathcal{D}, \mathcal{E}) = \frac{1}{2} \sum_{x \in X} |\mathcal{D}(x) - \mathcal{E}(x)|$, where X is the union of their support.

For a real number x and an integer η, we denote by $\lfloor x \rfloor_{2^{-\eta}}$, $\lceil x \rceil_{2^{-\eta}}$, $\lceil x \rfloor_{2^{-\eta}}$ the rounding of x down, up, or to the nearest number with precision $2^{-\eta}$, respectively. Namely, $\lfloor x \rfloor_{2^{-\eta}}$ is the largest number of the form $i/2^{\eta}$ (with i an integer) which is not larger than x, and similarly $\lceil x \rceil_{2^{-\eta}}$ is the smallest number of this form which is not smaller than x, and $\lceil x \rfloor_{2^{-\eta}}$ is one of the above which is closer to x (breaking ties arbitrarily). Omitting the $2^{-\eta}$ parameter means rounding to an integer (same as using 2^0).

2.1 Channels and Error Correcting Codes

A communication channel with input set $\mathcal{X}$ and output set $\mathcal{Y}$ is a transform that maps each input symbol $x \in \mathcal{X}$ to a distribution over the output symbols $\mathcal{Y}$. In this work we deal with additive-noise channels where $\mathcal{X} = \mathcal{Y}$ is an additive group, and the channel just adds to its input some random noise, chosen from a known distribution $\mathcal{D}$. Namely, $\mathsf{Ch} : x \mapsto x + \mathcal{D}$.

[2] Those compilers essentially construct a garbled circuit for the threshold function, with the secret being the output label. Hence, they require a very large public information, namely the garbled circuit itself.

We assume a memoryless channel: when sending a sequence of symbols, each symbol is transformed according to the channel Ch independently of the others (and their order is maintained).

An error-correction scheme is meant to facilitate reliable transmission of a sequence of symbols $m \in \mathcal{X}^k$ (for some k) over the channel Ch. For any input length k it consists of a code, defined by an encoding $\mathsf{Enc} : \mathcal{X}^k \to \mathcal{X}^n$ that adds redundancy, mapping the information sequence m to a longer code-word $w \in \mathcal{X}^n$ that will be sent over the channel, and by a matching decoding routine $\mathsf{Dec} : \mathcal{X}^n \to \mathcal{X}^k$ that attempts to recover the original information from the received sequence $\mathsf{Ch}(w)$. An error-correction scheme is a sequence of codes for increasing k.

The rate of a code is k/n, and the channel capacity is the highest possible rate (asymptotically as $k \to \infty$) of any scheme that achieves vanishing decoding error probability. For additive noise channels with noise distribution $\mathcal{D}$, the channel capacity is $1 - h(\mathcal{D})$ where h is the Shannon entropy function. In particular, for any channel Ch and any $\nu > 0$, there exist schemes with rate ν away from capacity (perhaps with inefficient encoding/decoding), in which the decoding error probability is bounded below $2^{-\Theta(n \cdot \nu^2)}$.

In this work we will be concerned with *Binary Symmetric Channels* (BSC, see Section 5) and *Additive White Gaussian Noise* channels (AWGN, see Appendix A). For those channels, there exist schemes with efficient encoding/decoding procedures that approach capacity and achieve vanishing error probability. (The dependence on the slackness parameter $\nu =$ capacity-minus-rate, affects the parameters that our blueprint can achieve, and will be discussed in the sequel.)

The "more noisy" relation

We say that a channel Ch' is more noisy than another channel Ch (or Ch is less noisy than Ch'), and denote $\mathsf{Ch} \preceq \mathsf{Ch}'$ or $\mathsf{Ch}' \succeq \mathsf{Ch}$, if there is some transform T such that $\mathsf{Ch}' = T(\mathsf{Ch})$. An example is when Ch' is obtained from Ch by adding more noise, $\mathsf{Ch}'(x) = \mathsf{Ch}(x) + \mathcal{D}$ for some noise distribution $\mathcal{D}$. It is easy to see that the capacity of Ch is at least as high as that of Ch'. Moreover, any error-correction scheme for Ch' also works for Ch.[3]

2.2 Wiretap Channel Transmission Schemes

A wiretap channel is a pair of communication channels $(\mathcal{R}, \mathcal{A})$ with the same input and output sets $\mathcal{X}, \mathcal{Y}$, where $\mathcal{R}$ is a channel from the sender to an intended receiver and $\mathcal{A}$ is the wiretap that goes to the adversary. Given a message m that the sender wants to send to the receiver, the goal is to encode it as $w = \mathsf{Enc}(m)$, so that m can be recovered (whp) from $\mathcal{R}(w)$, but not from $\mathcal{A}(w)$.

Bellare et al. defined in [6] the notion of semantically secure encryption scheme for a wiretap channel (that we prefer to call a *transmission scheme*[4]). The following is essentially their definition of distinguishing security. In our setting, it is sufficient to work with what they call a "seeded" scheme, where encoding and decoding depend on a public random seed.

[3] In theory, to use a decoder for Ch' we may need to apply T to the output of $\mathsf{Ch}(w)$ before we can decode it. In practice, decoders for the high-noise Ch' always work as-is also for the low-noise Ch.

[4] This is a keyless scheme, so it differs from cryptographic encryption.

▶ **Definition 1** (Secure Wiretap Transmission Schemes). *Let $(\mathcal{R}, \mathcal{A})$ be a wiretap channel (for message space $\mathcal{M}$), a secure transmission scheme for it consists of (seed-dependent*[5]*) encoding and decoding procedures $\mathsf{Enc}_{\mathsf{sd}}, \mathsf{Dec}_{\mathsf{sd}}$ such that*

Correctness. *For all $m \in \mathcal{M}$, $\Pr[\mathsf{Dec}_{\mathsf{sd}}(\mathcal{R}(\mathsf{Enc}_{\mathsf{sd}}(m))) = m] \geq 1 - \mathsf{negl}(|\mathsf{sd}|)$,*

Secrecy. *For all $m, m' \in \mathcal{M}$, $SD\left((\mathsf{sd}, \mathcal{A}(\mathsf{Enc}_{\mathsf{sd}}(m))), (\mathsf{sd}, \mathcal{A}(\mathsf{Enc}_{\mathsf{sd}}(m')))\right) \leq \mathsf{negl}(|\mathsf{sd}|)$,*

where the probability is over the channel randomness as well as the random selection of the seed sd, and negl is some negligible function.

The literature contains many constructions of wiretap channel schemes from error-correcting schemes, some of which we will be using in Section 5 and Appendix A. For the abstract blueprint that we present in Section 4, we need the "obvious" property of all the schemes in the literature, where if they work for one wiretap channel then they also work for all "easier channels." Namely, they are monotone in terms of the more-noisy relation:

▶ **Definition 2** (Monotone Schemes). *A secure transmission scheme $(\mathsf{Enc}, \mathsf{Dec})$ for a channel $(\mathcal{R}, \mathcal{A})$ is* noise-monotone *if it is also a secure transmission scheme for any channel $(\mathcal{R}', \mathcal{A}')$ such that $\mathcal{R}' \preceq \mathcal{R}$ and $\mathcal{A} \preceq \mathcal{A}'$.*

Clearly, the secrecy condition of a transmission scheme is always monotone. The correctness condition is monotone as long as the decoding error of the underlying code is not increased by *reducing* the noise level of the channel (which is true for all coding schemes that we know of).

3 Weighted Secret Sharing

A secret-sharing scheme is a two-phase multi-party protocol for $N + 1$ parties, a dealer and N shareholders. In the dealing phase, the dealer has a secret input $\boldsymbol{s}$, and it outputs a share for each shareholder, and optionally also a public share. In the reconstruction phase, a subset of the shareholders collect all their shares (and the public share if any) and attempt to use them in order to reconstruct the secret.

Each secret-sharing schemes comes with an *access structure*, consisting of a collection of qualified subsets $\Gamma \subset 2^{[N]}$ that should be able to reconstruct the secret, and a collection of *unqualified* subsets $\Psi \subset 2^{[N]}$ that should not be able to learn anything about the secret.[6] Non-perfect realizations of secret sharing come with a security parameter λ that is given as input to all the parties, and we require that the imperfections are negligible in λ.

Below we denote by $\mathsf{View}_S(\boldsymbol{s})$ the view of a subset of the shareholders $S \subset [N]$ when the secret $\boldsymbol{s}$ is shared, consisting of their own shares and the public share (if any). For a qualified set S we also denote by $\mathsf{Recover}(\mathsf{View}_S(\boldsymbol{s}))$ the value that these shareholders compute when trying to recover the secret.

▶ **Definition 3** (Secret Sharing). *A secret-sharing scheme for the access structure (Γ, Ψ) and the space of secrets $\mathcal{S}$, satisfies the following (for some negligible function $\mathsf{negl}(\cdot)$):*

Correctness. *For any qualified subset $S \in \Gamma$ and any secret $\boldsymbol{s} \in \mathcal{S}$,*

$$\Pr[\mathsf{Recover}(\mathsf{View}_S(\boldsymbol{s})) = \boldsymbol{s}] \geq 1 - \mathsf{negl}(\lambda).$$

[5] We use the seed length as the security parameter for this definition.

[6] Sometimes we have $\Psi = \overline{\Gamma}$, but rump schemes have $\Psi \subsetneq \overline{\Gamma}$.

Parameters: Weights $w_1, w_2, \ldots, w_N \in \mathbb{N}$, thresholds $0 < \alpha < \beta < 1$. Let $W := \sum_{i\in[N]} w_i$.

Sharing a secret $\boldsymbol{s} \in \{0,1\}^k$: Let $r = \lceil(\beta-\alpha)W\rceil$ and $k' = \lceil k/r\rceil$.

1. Break the secret into r chunks of length $\le k'$, let $\vec{s} \in (\{0,1\}^{k'})^r$ be the resulting vector;
2. Share $\vec{s}$ using $(\alpha W, \beta W; r, W)$ multi-secret sharing, party, $j \in [N]$, gets w_j shares.

Reconstructing the secret by a qualified set S:

3. Use multi-secret reconstruction with all revealed shares to recover $\vec{s}$;
4. Concatenate the entries of $\vec{s}$ to get $\boldsymbol{s}$.

Figure 1 A rate-efficient (α, β)-ramp WSS from multi-secret sharing.

Secrecy. *For any unqualified subset $T \in \Psi$ and any two secrets $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$, the view of T when sharing $\boldsymbol{s}$ is statistically close to the view when sharing $\boldsymbol{s}'$,*

$$SD\left(\mathsf{View}_T(\boldsymbol{s}), \mathsf{View}_T(\boldsymbol{s}')\right) \le \mathsf{negl}(\lambda).$$

In this work we study a relaxation of threshold weighted secret sharing, (α, β)-ramp weighted secret sharing.

▶ **Definition 4** ((α,β)-ramp weighted secret sharing)**.** *A (α, β)-ramp weighted secret sharing for $0 < \alpha < \beta < 1$, N shareholders, and weights $w_1, \ldots, w_N \in \mathbb{N}$, is a secret-sharing scheme for the access structure*

$$\Gamma = \{S \subseteq [N] : \sum_{i\in S} w_i \ge \beta W\} \text{ and } \Psi = \{T \subseteq [N] : \sum_{i\in T} w_i < \alpha W\},$$

where $W = \sum_{i\in[N]} w_i$.

Below we often use the notation $\epsilon = \beta - \alpha$ when discussing the parameters of ramp WSS schemes.

3.1 Ramp WSS from Multi-Secret Sharing

A $(T_1, T_2; r, N)$ multi-secret sharing scheme shares r secrets (from some domain) among N shareholders, with secrecy when T_1 or less of the shares are revealed and recovery when T_2 or more shares are revealed. A packed Shamir sharing, where multiple secrets are encoded in different evaluation points of a degree-$(T-1)$ polynomial, yields a $(T-r, T; r, N)$ multi-secret sharing scheme over any field of size $\ge N + r$, where each share is only a single field element. Hence it achieves a "rate" of $|\text{secret}|/|\text{share}| = r$.

This can be converted to a ramp WSS scheme using the obvious approach of giving w shares to a weight-w shareholder. This construction is described in Figure 1. To get an (α, β)-ramp WSS we need a multi-secret scheme with $N := W$, $T_1 := \alpha W$, and $T_2 := \beta W$. Using the above construction, we can pack $r = T_2 - T_1 = \epsilon W$ field elements while each underlying share is a single element.

Each shareholder in the resulting WSS scheme holds at most W shares of the underlying scheme, so we get a WSS scheme with share size $\le W$ element that can handle secrets of size upto ϵW elements. This yields encoding rate of

$$|\text{secret}|/|\text{share}| \ge \frac{\epsilon W}{W} = \epsilon,$$

Parameters: Weights $w_1, w_2, \ldots, w_N \in \mathbb{N}$, thresholds $0 < \alpha < \beta < 1$. Let $W := \sum_{i\in[N]} w_i$.

1. Let $\eta := \left\lceil \log \frac{5N}{\beta-\alpha} \right\rceil$. For all $j \in [N]$, set $w'_j := 2^\eta \cdot \left\lceil \frac{w_j}{W} \right\rceil_{2^{-\eta}}$.
2. Use the Ramp WSS from Figure 1 with the w'_j's and thresholds $\alpha' = \alpha + \frac{\beta-\alpha}{5}$ and $\beta' = \beta - \frac{\beta-\alpha}{4}$.

Figure 2 A rounded (α, β)-ramp weighted secret sharing.

as long as the secret is long enough (i.e., at least ϵW field elements). This scheme is not very useful for short secrets, however, as its efficiency depends on breaking the secret into many chunks. In particular, the size of shares is still W (or more) in the worst case, regardless of how small is the secret. [7]

3.2 A Rounding-Based (α, β)-ramp WSS Protocol

We note that simple rounding can be used to roughly replace the dependence on W in the above scheme by dependent on N/ϵ. Specifically, we use the construction from Figure 1 to implement a modified version of the system, with weights that are rounded to precision of only about $(\beta - \alpha)/N$. Due to rounding errors, the modified version has a smaller gap $\epsilon' < \beta - \alpha$, but the increase can be controlled by setting the precision appropriately. Specifically, with precision of $(\beta - \alpha)/5N$ we can get $\epsilon' \geq \epsilon/2$. This simple protocol is described in Figure 2.

▶ **Lemma 5.** *The protocol outline in Figure 2 is an (α, β)-ramp weighted secret sharing scheme.*

Proof. By our choice of η we get $N/2^\eta \leq (\beta - \alpha)/5$, and for every set $J \subseteq [N]$ we have

$$2^\eta \sum_{j\in J} w_j/W \;\leq\; \sum_{j\in J} w'_j \;<\; |J| + 2^\eta \sum_{j\in J} w_j/W.$$

In particular for $J = [N]$ we have $W' = \sum_{j\in[N]} w'_j \in [2^\eta, 2^\eta + N]$. For any non-qualified set $J \subseteq [N]$ with $\sum_{j\in J} w_j \leq \alpha W$ we therefore have

$$\sum_{j\in J} w'_j/W' \leq \frac{N + 2^\eta \sum_{j\in J} w_j/W}{2^\eta} \leq \frac{N + 2^\eta \cdot \alpha}{2^\eta} \leq (\beta-\alpha)/5 + \alpha = \alpha'.$$

Similarly, for any qualified set $J \subseteq [N]$ with $\sum_{j\in J} w_j \geq \beta W$ we have

$$\sum_{j\in J} w'_j/W' \geq \frac{2^\eta \sum_{j\in J} w_j/W}{N + 2^\eta} \geq \frac{\beta}{1 + (N/2^\eta)} \geq \frac{\beta}{1 + (\beta-\alpha)/5} \overset{(*)}{\geq} \beta - (\beta-\alpha)/4 = \beta'.$$

To see why inequality $(*)$ holds, note that

$$\frac{\beta}{1+(\beta-\alpha)/5} = \frac{\beta(1+(\beta-\alpha)/5)}{1+(\beta-\alpha)/5} - \frac{\beta(\beta-\alpha)/5}{1+(\beta-\alpha)/5} = \beta - \frac{\beta(\beta-\alpha)}{5-(\beta-\alpha)} \geq \beta - \frac{\beta-\alpha}{4}. \blacktriangleleft$$

In terms of performance for the protocol of Figure 2, the number of shares a party can receive is upper-bounded by $W' < N + 2^\eta \leq N\big(1 + \frac{10}{\beta-\alpha}\big)$. Hence, the size of shares in this scheme grows with $O(N/\epsilon)$ instead of the total weight W.

[7] In other contexts it is sometimes helpful to use algebraic-geometric codes instead of the Reed-Solomon codes of Shamir sharing, as it enables the use of smaller fields. In our case this does not seem to help, since the inefficiency comes from the number of field elements and not their size.

Sharing a secret $\boldsymbol{s} \in \{0,1\}^k$, with security parameter λ:

1. If the wiretap scheme is seeded, choose a random seed sd of length λ;

2. $\forall j \in [N]$, draw $\boldsymbol{e}_j \leftarrow \mathcal{D}_{w_j}$ and send to party j;

3. Publish sd and $\boldsymbol{g} = Enc_{\mathsf{sd}}(\boldsymbol{s}) + \sum_{j\in[N]} \boldsymbol{e}_j$.

Reconstructing the secret by a qualified set S:

Set $\boldsymbol{g}' = \boldsymbol{g} - \sum_{j\in S} \boldsymbol{e}_j$ and output $\mathsf{Dec}_{\mathsf{sd}}(\boldsymbol{g}')$.

Figure 3 The generic framework for ramp weighted secret sharing from wiretap channels.

4 A Blueprint for WSS from Wiretap Channels

Let $w_1, \ldots, w_N$ be the concrete weights that we want to implement and $0 < \alpha < \beta < 1$ be the parameters that we want to achieve. Denote $W = \sum_{i\in[N]} w_i$. An instance of our blueprint operates in some additive group $\mathcal{X}$, and consists of two components:

- A mapping from weights $w \in \mathbb{N}$ to noise distributions $\mathcal{D}_w$ over $\mathcal{X}$.
- A (seeded) noise-monotone secure transmission scheme $(\mathsf{Enc}, \mathsf{Dec})$ for a wiretap channel $(\mathcal{R}, \mathcal{A})$ (cf. Definition 1), such that:
    - For any qualified subset $S \subseteq [N]$ with $\sum_{i\in S} w_i \geq \beta W$, the channel $\mathcal{R}$ is more noisy than adding all the noise distributions *outside* S. Namely, $C_S \preceq \mathcal{R}$ where $C_S : x \mapsto x + \sum_{i\notin S} \mathcal{D}_{w_i}$.
    - For any unqualified subset $T \subseteq [N]$ with $\sum_{i\in S} w_i \leq \alpha W$, the channel $\mathcal{A}$ is less noisy than adding all the noise distributions *outside* T. Namely, $C_T \succeq \mathcal{A}$ where $C_T : x \mapsto x + \sum_{i\notin T} \mathcal{D}_{w_i}$.

Given these components, our WSS scheme is described in Figure 3.

▶ **Lemma 6.** *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is a noise-monotone secure transmission scheme for a wiretap channel* $(\mathcal{R}, \mathcal{A})$*, as per Definitions 1 and 2, that satisfy the conditions above. Then the scheme from Figure 3 is a secure* (α, β)*-ramp weighted secret-sharing scheme.*

Proof. This holds more or less by definition. Consider an arbitrary qualified set S and an arbitrary unqualified set T. Then by construction we have $C_S \preceq \mathcal{R}$ and $\mathcal{A} \preceq C_T$, and since $(\mathsf{Enc}, \mathsf{Dec})$ is noise-monotone then it is also a secure transmission scheme for the wiretap channel (C_S, C_T). This means on one hand that for the qualified set S, seeing $y = C_S(\mathsf{Enc}(\boldsymbol{s}))$, we have $\mathsf{Dec}(y) = \boldsymbol{s}$ with all but negligible probability. On the other hand, the unqualified set T, seeing only $C_T(\mathsf{Enc}(\boldsymbol{s}))$, cannot distinguish it from $C_T(\mathsf{Enc}(\boldsymbol{s}'))$ except with a negligible advantage. ◀

The public share

Our solutions, as well as some solutions from the literature (such as [4]), use a public share, which is known to everyone, in addition to the individual shares of the shareholders. Clearly, it is possible to eliminate the public share by adding it to each individual share, and in our case this will at most double the share size of parties. In some solutions in the literature, however, the public share is much larger than the individual shares. Here we chose to account for the public share separately and only count it once (rather than once per shareholder).

5 Constructions from Binary Symmetric Wiretap Channels

5.1 Background

5.1.1 Binary Symmetric Channels

A binary symmetric channel (BSC) is used for sending bits. It is associated with a "crossover probability" $p \leq 1/2$, which is the probability that the received bit differs from the one that was sent. Namely, we have a Bernoulli error variable B_p with $\Pr[B_p = 1] = p$ and $\Pr[B_p = 0] = 1 - p$, and the channel is defined on message space $\{0, 1\}$ as $BSC_p : x \mapsto x + B_p \bmod 2$.

The capacity of BSC_p is $1 - h(p)$, where h is the binary entropy function. If we are not concerned with efficient decoding, then random linear codes (with ML/MAP decoding) have rates that approach the channel capacity with exponentially small error probability. Capacity-approaching constructions with efficient decoding are known using concatenated codes [17] or polar codes [1, 18], with somewhat weaker bounds on the decoding error. For example, [2, 19] show that the error probability for block-length n and rate $1 - h(p) - \nu$ is at most $\exp(-\Theta(\sqrt{n}))$, where the constant in the exponent depends on p and ν. Later results feature stronger bounds in terms of the block-length n with polynomial dependence on the slackness ν. In particular, we have

▶ **Lemma 7** (Corollary of [11], Thm 17). *For any $p < 1/2$, $\nu < 1-h(p)$, and $\mu < 1$, there exists a code for BSC_p with rate $1 - h(p) - \nu$, block length $n = \mathsf{poly}_\mu(1/\nu)$ (for some polynomial that depends on μ), error probability $\exp(-n^\mu)$, and decoding complexity $O(n \log n)$.*

It is known that the polynomial dependence on $1/\nu$ is quadratic for any discrete memoryless channel, while for some efficient constructions there is evidence that $\text{poly}_{1/2}(x) \leq x^{4.6}$ [23, 28].

5.1.2 Wiretap Schemes for Binary Symmetric Channels

Bellare et al. also described in [6] a construction called ItE (Invert-then-Encode) for discrete wiretap channels, building on error-correction. The construction realizes Definition 1 for the channels $(\mathcal{R}, \mathcal{A})$, using a code with low decoding error probability for $\mathcal{R}$, at a rate noticeably larger than the capacity of $\mathcal{A}$. (In particular, if the code rate approaches the capacity of $\mathcal{R}$ then this construction approaches the secrecy capacity of the wiretap channel.)

The ItE construction has integer parameters $b < k < n$ (with values as set later in this section). Identifying $\{0, 1\}^k$ with the finite field $\mathbb{F}_{2^k}$, this is a seeded construction with seed space the multiplicative group $\mathbb{F}_{2^k} \setminus \{0^k\}$ and message space $\{0, 1\}^b$. In addition, it uses error-correction encoding $\mathsf{Enc} : \{0, 1\}^k \to \{0, 1\}^n$ and the corresponding decoding $\mathsf{Dec} : \{0, 1\}^n \to \{0, 1\}^k$. The encoding and decoding routines of the ItE construction (denoted $\mathsf{Enc}'_{\mathsf{sd}}, \mathsf{Dec}'_{\mathsf{sd}}$) are described in Figure 4. The following is a re-phrasing of Lemma 5.3 and Lemmas 5.5-5.6 from [5]:

▶ **Lemma 8** ([5], Lemma 5.3). *If $(\mathsf{Enc}, \mathsf{Dec})$ is an error-correction scheme with decoding-error probability at most ϵ for the channel $\mathcal{R}$, then the ItE scheme $(\mathsf{Enc}_{\mathsf{sd}}, \mathsf{Dec}_{\mathsf{sd}})$ from Figure 4 is correct for $(\mathcal{R}, \mathcal{A})$ with correctness holding with probability $\geq 1 - \epsilon$.* ◀

▶ **Lemma 9** (Corollary of [5], Lemmas 5.5-5.6). *Let $\mathcal{A}$ be a symmetric memoryless channel with capacity $c(\mathcal{A})$. Assume that $\frac{k}{n}$ (the rate of Enc) is larger than $c(\mathcal{A})$, denote the slackness by $\rho = \frac{k}{n} - c(\mathcal{A})$, and let λ be the security parameter. Then for any $0 < \delta < \rho - \frac{2\lambda}{n}$, setting $b := \lfloor n(\rho - \delta) - 2\lambda - 2 \rfloor$ in the ItE construction yields a wiretap transmission scheme with secrecy upto statistical distance $4 \cdot 2^{-\delta^2 n/11} + 2 \cdot 2^{-\lambda}$.*

```
Encoding: $\mathsf{Enc}'_{\mathsf{sd}}(M \in \{0,1\}^b)$ with seed $\mathsf{sd} \in \mathbb{F}_{2^k} \setminus \{0^k\}$:
1. Choose a random $R \leftarrow \{0,1\}^{k-b}$, let $Y = (M|R) \in \mathbb{F}_{2^k}$ be the concatenation;
2. Set $X := Y/\mathsf{sd} \in \mathbb{F}_{2^k}$;
3. Send the message $W := \mathsf{Enc}(X) \in \{0,1\}^n$.
Decoding: $\mathsf{Dec}'_{\mathsf{sd}}(W' \in \{0,1\}^n)$:
1. Use error-correction to get $X' = \mathsf{Dec}(W')$;
2. Compute $Y' := X' \cdot \mathsf{sd}$ ;
3. Output $M'$, the first $b$ bits of $Y'$.
```

Figure 4 The ItE construction from [6].

Plugging the coding parameter from above, we get the following instantiation:

▶ **Corollary 10.** *For a binary symmetric wiretap channel (BSC_{p_R}, BSC_{p_A}) with $0 \le p_R < p_A < 1/2$, denote $\xi := h(p_A) - h(p_R)$. There exists an instance of the ItE scheme $(\mathsf{Enc}_{\mathsf{sd}}, \mathsf{Dec}_{\mathsf{sd}})$ with security parameter λ and*

- *Encoding size $n = \max\big(\mathsf{poly}_{\frac{1}{2}}(\frac{4}{\xi}), \lambda^2, \frac{44\lambda}{\xi^2}\big)$;* [8]
- *Seed space $\mathbb{F}_{2^k} \setminus \{0^k\}$ with $k = (1 - h(p_A) + \frac{3\xi}{4})n = (1 - h(p_R) - \frac{\xi}{4})n$; and*
- *Message space $\{0,1\}^b$, $b \ge (\frac{\xi}{4} - \frac{2}{\lambda})n - 2$;*

such that

- *For all $m \in \{0,1\}^b$, $\Pr[\mathsf{Dec}_{\mathsf{sd}}(BSC_{p_R}(\mathsf{Enc}_{\mathsf{sd}}(m))) = m] \ge 1 - 2^{-\lambda}$;*
- *For all $m, m' \in \{0,1\}^b$,*

$$SD\big((\mathsf{sd}, BSC_{p_A}(\mathsf{Enc}_{\mathsf{sd}}(m))),\ (\mathsf{sd}, BSC_{p_A}(\mathsf{Enc}_{\mathsf{sd}}(m')))\big) \le 6 \cdot 2^{-\lambda}.$$

Proof. Recall that k determines both the seed space of the ItE construction and the input space for the underlying error-correcting code. The rate of the underlying code is therefore $k/n = 1 - h(p_R) - \frac{\xi}{4}$, and by Lemma 7 we can find such codes as soon as the encoding-length exceeds $\mathsf{poly}_{1/2}(4/\xi)$, with decoding error probability at most $\exp(-n^{1/2}) < 2^{-\sqrt{n}}$. If $n \ge \lambda^2$ then this is bounded below $2^{-\lambda}$, and due to Lemma 8 the same holds for correctness of the ItE construction.

For the secrecy part, we have rate $k/n = 1 - h(p_A) + \frac{3\xi}{4}$, and we use $\delta = \frac{\xi}{2}$ in Lemma 9. This yields $b = \frac{\xi}{4}n - 2\lambda - 2$, and since $n \ge \lambda^2$ then $b \ge (\frac{\xi}{4} - \frac{2}{\lambda})n - 2$. If we also have $n \ge \frac{44\lambda}{\xi^2}$, then $\delta^2 n/11 \ge (\xi/2)^2 \cdot (44\lambda/\xi^2)/11 = \lambda$, and therefore the statistical distance is bounded by

$$4 \cdot 2^{-\delta^2 n/11} + 2 \cdot 2^{-\lambda} \le 4 \cdot 2^{-\lambda} + 2 \cdot 2^{-\lambda} = 6 \cdot 2^{-\lambda}. \qquad \blacktriangleleft$$

5.1.2.1 Remark

Different from most works in the literature, in the setting above we do not aim at achieving the secrecy capacity in the limit. Rather, we try to maintain a small encoding size n relative not just to the message size b, but also to the security parameter λ and the parameters p_R, p_A. [9]

[8] $\mathsf{poly}_{\frac{1}{2}}$ is the polynomial from Lemma 7 for $\mu = \frac{1}{2}$.

[9] In particular, we opted for losing a constant factor in the ratio b/n in return for better dependency on λ and ξ.

Parameters:

Weights $w_1, w_2, \ldots, w_N \in \mathbb{N}$, thresholds $0 < \alpha < \beta < 1$, security parameter λ.

- Let $W := \sum_{i \in [N]} w_i$ and $\gamma := \frac{0.4}{1-\alpha}$.
- Denote $g(x) := \frac{1-\exp(-x)}{2}$, and let $p_R := g(\gamma(1-\beta))$ and $p_A := g(\gamma(1-\alpha))$.
- Let $(\mathsf{Enc}, \mathsf{Dec})$ (with parameters n, k, b) be as in the ItE construction from Corollary 10 for the wiretap channel (BSC_{p_R}, BSC_{p_A}).

Sharing a secret $\boldsymbol{s} \in \{0,1\}^k$:

1. $\forall j \in [N]$, set $p_j := g(\frac{\gamma \cdot w_j}{W})$, draw $\boldsymbol{e}_j \leftarrow (\mathsf{Bernoulli}_{p_j})^n$ and send to party j;
2. Draw a random $\mathsf{sd} \in \mathbb{F}_{2^k} \setminus \{0^k\}$, publish sd and $\boldsymbol{g} := \mathsf{Enc}_{\mathsf{sd}}(\boldsymbol{s}) + \sum_{j \in [N]} \boldsymbol{e}_j \bmod 2$.

Reconstructing the secret by a qualified set S:

Set $\boldsymbol{g}' = \boldsymbol{g} + \sum_{j \in S} \boldsymbol{e}_j \bmod 2$ and output $\mathsf{Dec}_{\mathsf{sd}}(\boldsymbol{g}')$.

Figure 5 Weighted secret sharing from symmetric binary wiretap channels.

5.2 Our Construction

In Figure 5 we show how to use the ItE instance from Corollary 10 to get an (α, β)-ramp WSS for given weights $w_1, w_2, \ldots, w_N$ and thresholds $0 < \alpha < \beta < 1$.

Clearly, this construction is an instance of the blueprint from Figure 3, instantiated over the additive group $\mathbb{F}_{2^k}$, using the noise distributions $D_w = \mathsf{Bernoulli}_{g(\gamma \cdot w/W)}$ and the ItE construction from Corollary 10 for the wiretap channel (CSB_{p_R}, CSB_{p_A}). It is also clear that the ItE construction is noise-monotone (since the underlying error-correction codes are).

The only thing left to prove in order to use Lemma 6, is that for any qualified S and unqualified T, the corresponding channels satisfy $C_S \preceq BSC_{p_R}$ and $C_T \succeq BSC_{p_A}$. To that end, we use the following technical lemma:

▶ **Lemma 11.** *Let $\mathcal{B}_1, \ldots, \mathcal{B}_t$ be independent Bernoulli random variables with* $\Pr[\mathcal{B}_j = 1] = \frac{1-\exp(-u_j)}{2}$*, and denote $\mathcal{S} := \sum_{j \in [t]} \mathcal{B}_j \bmod 2$ then $\mathcal{S}$ is a Bernoulli random variable with:*

$$\Pr[\mathcal{S} = 1] = \frac{1 - \exp(-\sum_{j \in [t]} u_j)}{2}.$$

Proof. We prove the lemma by induction on t. The base case, where $t = 1$, is trivial. For $t > 1$, we have that

$$\begin{aligned}
\Pr[\mathcal{S} = 1] &= \Pr\left[\sum_{j \in [t-1]} \mathcal{B}_j \bmod 2 = 1 \ \&\ \mathcal{B}_t = 0\right] + \Pr\left[\sum_{j \in [t-1]} \mathcal{B}_j \bmod 2 = 0 \ \&\ \mathcal{B}_t = 1\right] \\
&= \frac{1 - \exp\left(\sum_{j \in [t-1]} u_j\right)}{2} \cdot \frac{1 + \exp(x_t)}{2} + \frac{1 + \exp\left(\sum_{j \in [t-1]} u_j\right)}{2} \cdot \frac{1 - \exp(x_t)}{2} \\
&\qquad \text{(by the inductive hypothesis)} \\
&= \frac{1 - \exp\left(-\sum_{j \in [t]} u_j\right)}{2}.
\end{aligned}$$

Thus, the inductive step holds. ◀

We can now complete the proof that the ItE-based construction above satisfies all the conditions of Lemma 6.

▶ **Corollary 12.** *With the parameters as set in Figure 5 and a straightforward application of Lemma 11:*

(A) *For every subset $S \subseteq [N]$ with $\sum_{j\in S} w_j \geq \beta W$, we have $C_S \preceq BSC_{p_R}$ where $C_S : x \mapsto x + \sum_{j\notin S} \mathsf{Bernoulli}_{p_j}$.*

(B) *For every subset $T \subseteq [N]$ with $\sum_{j\in S} w_j \leq \alpha W$, we have $C_T \succeq BSC_{p_A}$ where $C_T : x \mapsto x + \sum_{j\notin T} \mathsf{Bernoulli}_{p_j}$.*

An Alternative Presentation

An alternative way of describing the scheme from Figure 5, is that the noise component for party j with weight w_j is set as the sum (modulo 2) of w_j IID random vectors, all of the form $\left(\mathsf{Bernoulli}_{g(\gamma/W)}\right)^n$. By Lemma 11, this noise vector indeed has the form $\left(\mathsf{Bernoulli}_{p_j}\right)^n$, where $p_j = g(\gamma \cdot w_j/W)$.

5.3 Performance Characteristics of This Construction

Let $\epsilon = \beta - \alpha > 0$ and $h : p \mapsto -p \log p - (1-p)\log(1-p)$ the binary entropy function (recall that log is in base 2). To get the best parameters from Corollary 10, we want to set the parameter γ so as to maximize $\xi := h(p_A) - h(p_R)$, where:

$$p_A = g(\gamma(1-\alpha)), \qquad p_R = g(\gamma(1-\beta)), \qquad \text{recalling that } g : x \mapsto \frac{1-\exp(-x)}{2}.$$

Denote the function $f : x \mapsto h(g(x))$. The mean value theorem implies that:

$$\begin{aligned}\xi &= f(\gamma(1-\alpha)) - f(\gamma(1-\beta)) && (1)\\ &\geq (\gamma(1-\alpha) - \gamma(1-\beta)) \cdot \inf_{\gamma(1-\beta)<x<\gamma(1-\alpha)} f'(x) \\ &= \epsilon \cdot \gamma \cdot \inf_{\gamma(1-\beta)<x<\gamma(1-\alpha)} f'(x). && (2)\end{aligned}$$

where f' is the derivative of f.

Let us now compute f'. We have $h'(p) = \log(1/p - 1)$ and $g'(x) = \exp(-x)/2$. Thus:

$$f'(x) = \frac{1}{2}\exp(-x) \cdot \log\left(\frac{2}{1-\exp(-x)} - 1\right).$$

We remark that f' is decreasing, because $\exp(-x)$ is decreasing and $\log\left(\frac{2}{1-\exp(-x)} - 1\right)$ is decreasing. Therefore Equation (2) implies:

$$\xi \geq \epsilon \cdot \gamma \cdot f'(\gamma(1-\alpha)) = \frac{\epsilon}{1-\alpha} \cdot \gamma' \cdot f'(\gamma') \geq \epsilon \cdot \gamma' \cdot f'(\gamma')$$

where $\gamma' := \gamma(1-\alpha)$.

To maximize our lower bound of ξ, we just need to maximize $\gamma' \cdot f'(\gamma')$. The optimal γ' is about 0.4. In particular, setting $\gamma' = 0.4$ gives $\gamma' \cdot f'(\gamma') \geq 0.31$. Thus we can set $\gamma = \frac{0.4}{1-\alpha}$, which implies

$$\xi = h(p_A) - h(p_R) \geq 0.31 \cdot \epsilon. \qquad (3)$$

By Corollary 10, there is a transmission scheme $(\mathsf{Enc_{sd}}, \mathsf{Dec_{sd}})$ for the wiretap channel (BSC_{p_R}, BSC_{p_A}) with correctness/secrecy upto $O(2^{-\lambda})$ and parameters

- **Encoding length:** $n \leq \max\left(\mathsf{poly}_{\frac{1}{2}}(\frac{13}{\epsilon}), \lambda^2, \frac{458\lambda}{\epsilon^2}\right)$;
- **Message length:** $b \geq (\frac{\epsilon}{13} - \frac{2}{\lambda})n - 2$;
- **Seed length:** $k = \lceil (1 - h(p_A) + \frac{\epsilon}{4})n \rceil$.

Recall that for this scheme, we have secrets of length b, each shareholder gets a share of length n, and the public share is of size $n+k$. Note also that n, k, b depend only the thresholds α, β and not on the weights themselves. Thus, we get a scheme where the share sizes are independent of the weights, and the rate is $b/(2n+k) = \Theta(\epsilon)$.

When the gap $\epsilon = \beta - \alpha$ is a constant, we can obtain this constant rate already for constant-size secret. As the gap gets smaller, the share sizes grow as a polynomial in $1/\epsilon$, so we can only get $\Theta(\epsilon)$ rate for longer secrets. For example, assuming that constructions such as [28] yield good binary codes with efficient decoding and $\mathsf{poly}_{\frac{1}{2}}(x) = x^{4.6}$, we get $n \approx (13/\epsilon)^{4.6}$.

6 Conclusions

In this work, we study a ramp weighted secret sharing, with a gap between the qualified and unqualified sets, and described two different types of constructions, one based on rounding and the other using a new connection to wiretap schemes. Both types have share size independent of total weight, and dependent only the gap between qualified and unqualified sets. We described in detail a construction based on binary symmetric wiretap channels, and sketched one based on AWGN. It may be interesting to explore other channels as well, to see if any of then can offer concrete parameter improvements.

References

1 Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009. doi:10.1109/TIT.2009.2021379.

2 Erdal Arikan and Emre Telatar. On the rate of channel polarization. In *2009 IEEE International Symposium on Information Theory*, pages 1493–1495, 2009. doi:10.1109/ISIT.2009.5205856.

3 Amos Beimel, Tamir Tassa, and Enav Weinreb. Characterizing ideal weighted threshold secret sharing. In *Theory of Cryptography Conference*, pages 600–619. Springer, 2005.

4 Amos Beimel and Enav Weinreb. Monotone circuits for monotone weighted threshold functions. *Information Processing Letters*, 97(1):12–18, 2006. doi:10.1016/j.ipl.2005.09.008.

5 Mihir Bellare and Stefano Tessaro. Polynomial-time, semantically-secure encryption achieving the secrecy capacity. *IACR Cryptology ePrint Archive*, page 22, 2012. URL: http://eprint.iacr.org/2012/022.

6 Mihir Bellare, Stefano Tessaro, and Alexander Vardy. Semantic security for the wiretap channel. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 294–311. Springer, 2012. Also available from https://arxiv.org/abs/1201.2205. doi:10.1007/978-3-642-32009-5_18.

7 Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988. doi:10.1007/0-387-34799-2_3.

8 Fabrice Benhamouda, Shai Halevi, and Lev Stambler. Weighted secret sharing from wiretap channels. *IACR Cryptol. ePrint Arch.*, page 1578, 2022. URL: https://eprint.iacr.org/2022/1578.

9 G. R. Blakley and Catherine A. Meadows. Security of ramp schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 242–268. Springer, 1984. doi:10.1007/3-540-39568-7_20.

10 George R. Blakley. Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318, 1979. doi:10.1109/MARK.1979.8817296.

11 Jaroslaw Blasiok, Venkatesan Guruswami, and Madhu Sudan. Polar codes with exponentially small error at finite block length. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018*, volume 116 of *LIPIcs*, pages 34:1–34:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.34.

12 Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing.* Cambridge University Press, 2015. URL: http://www.cambridge.org/de/academic/subjects/computer-science/cryptography-cryptology-and-coding/secure-multiparty-computation-and-secret-sharing?format=HB&isbn=9781107043053.

13 Claude Crépeau. Efficient cryptographic protocols based on noisy channels. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 306–317. Springer, 1997. doi:10.1007/3-540-69053-0_21.

14 Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 42–52. IEEE Computer Society, 1988. doi:10.1109/SFCS.1988.21920.

15 U. Erez and R. Zamir. Achieving 1/2 log(1+SNR) on the AWGN channel with lattice encoding and decoding. *IEEE Transactions on Information Theory*, 50(10):2293–2314, 2004. doi:10.1109/TIT.2004.834787.

16 Oriol Farras and Carles Padró. Ideal hierarchical secret sharing schemes. *IEEE transactions on information theory*, 58(5):3273–3286, 2012.

17 Venkatesan Guruswami and Atri Rudra. Concatenated codes can achieve list-decoding capacity. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 258–267, USA, 2008. Society for Industrial and Applied Mathematics.

18 Venkatesan Guruswami and Patrick Xia. Polar codes: Speed of polarization and polynomial gap to capacity. *IEEE Trans. Inf. Theory*, 61(1):3–16, 2015. doi:10.1109/TIT.2014.2371819.

19 S. Hamed Hassani, Ryuhei Mori, Toshiyuki Tanaka, and Rüdiger L. Urbanke. Rate-dependent analysis of the asymptotic behavior of channel polarization. *IEEE Transactions on Information Theory*, 59(4):2267–2276, 2013. doi:10.1109/TIT.2012.2228295.

20 Ling Liu, Yanfei Yan, and Cong Ling. Achieving secrecy capacity of the gaussian wiretap channel with polar lattices. *IEEE Transactions on Information Theory*, 64(3):1647–1665, 2018. Also available at https://arxiv.org/abs/1503.02313. doi:10.1109/TIT.2018.2794327.

21 Ling Liu, Yanfei Yan, Cong Ling, and Xiaofu Wu. Construction of capacity-achieving lattice codes: Polar lattices. *IEEE Transactions on Communications*, 67(2):915–928, 2019. Also available at https://arxiv.org/abs/1411.0187. doi:10.1109/TCOMM.2018.2876113.

22 Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. doi:10.1137/S0097539705447360.

23 Marco Mondelli, S. Hamed Hassani, and Rüdiger L. Urbanke. Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors. *IEEE Transactions on Information Theory*, 62(12):6698–6712, 2016. doi:10.1109/TIT.2016.2616117.

24 Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

25 Tamir Tassa. Hierarchical threshold secret sharing. *Journal of cryptology*, 20(2):237–264, 2007.

26 Himanshu Tyagi and Alexander Vardy. Explicit capacity-achieving coding scheme for the gaussian wiretap channel. In *2014 IEEE International Symposium on Information Theory*, pages 956–960, 2014. See also https://arxiv.org/abs/1412.4958. doi:10.1109/ISIT.2014.6874974.

27 Vinod Vaikuntanathan, Arvind Narayanan, K. Srinathan, C. Pandu Rangan, and Kwangjo Kim. On the power of computational secret sharing. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003*, volume 2904 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2003. doi:10.1007/978-3-540-24582-7_12.

28 Hsin-Po Wang, Ting-Chun Lin, Alexander Vardy, and Ryan Gabrys. Sub-4.7 scaling exponent of polar codes. arXiv 2204.11683, 2022. URL: https://arxiv.org/abs/2204.11683.

29 Xukai Zou, Fabio Maino, Elisa Bertino, Yan Sui, Kai Wang, and Feng Li. A new approach to weighted multi-secret sharing. In *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, 2011.

A Constructions from AWGN Wiretap Channels

Below we sketch an AWGN-based construction. It is plausible that such construction can provide somewhat better performance than BSC-based construction, since AWGN-code use "soft decoding" vs. the "hard decoding" that's inherent in BSC code. We do not know if the existing codes actually realize such improvement, however. Moreover, the construction below features logarithmic dependence on the number of parties.

A.1 Background

A.1.1 Additive White Gaussian Noise Channels

Additive white Gaussian noise channels (AWGN) communicate real numbers rather than bits. For each symbol $x \in \mathbb{R}$ transmitted over the channel, the received symbol is $y = x + e$ (addition over the reals), where e is a zero-mean Normal random variable. The variance σ^2 of e is the noise level of the channel.

Symbols transmitted over the channel are chosen subject to some power constraint, specifically their (expected) square is bounded by the *power parameter* P of the sender. The quality of the channel is determined by the ratio between the power and the noise, called the signal-to-noise ratio: $SNR = P/\sigma^2$. [10] Below it will be convenient to fix the power to $P = 1$ and set the variance accordingly. We denote the AWGN channel with variance σ^2 (and power $P = 1$) by $AWGN_{\sigma^2} : x \mapsto x + \mathcal{N}(0, \sigma^2)$. The capacity of this channel (denoted $c(\sigma)$ below) is

$$c(\sigma) := \mathsf{capacity}(AWGN_{\sigma^2}) = \ln\left(1 + \frac{1}{\sigma^2}\right).$$

(The general formula is $\ln\left(1 + \frac{P}{\sigma^2}\right)$ but we are fixing $P = 1$.) There are known constructions of error-correcting codes with efficient decoding for the AWGN that approach capacity, see for example [15, 21]. While AWGN codes can perhaps achieve somewhat better performance than BSC codes (since they use "soft decoding") this improvement has little effect on their asymptotic behavior. In particular, for slackness parameter $\nu < c(\sigma)$, there exist codes for $AWGN_{\sigma^2}$ with rate $c(\sigma) - \nu$, block length $n = \mathsf{poly}(1/\nu)$ (for some polynomial), error probability $\exp(-\sqrt{n})$, and decoding complexity polynomial in n.

[10] Clearly, scaling P and σ^2 by the same factor has no effect on the channel quality.

A.1.2 AWGN Wiretap Channels

Tyagi and Vardy described in [26] a modular construction (in the same spirit as [6]) that combines AWGN codes with randomness extractors. If the underlying code approaches the receiver channel capacity, then the Tyagi-Vardy scheme can be made to approach the secrecy capacity of the wiretap channel. A different approach for a secrecy-capacity-approaching schemes was provided by Liu et al. [20].

These AWGN constructions may be practically more efficient than their BSC counterparts, but as far as we know the improvement has little effect on their asymptotic behavior. Namely, for an AWGN wiretap channel $(AWGN_{\sigma_r^2}, AWGN_{\sigma_a^2})$ with $0 \leq \sigma_r < \sigma_a$, denote $\xi := c(\sigma_r) - c(\sigma_a)$. Then the constructions in [26, 20] provide seeded wiretap transmission schemes $(\mathsf{Enc}_{\mathsf{sd}}, \mathsf{Dec}_{\mathsf{sd}})$ with security parameter λ and

- Encoding size $n = \max\big(\mathsf{poly}(\frac{1}{\xi}), \lambda^2, O(\frac{\lambda}{\xi^2})\big)$;
- Seed size $k = (c(\sigma_a) + \Theta(\xi))n = (c(\sigma_r) - \Theta(\xi))n$; and
- Message space $\{0,1\}^b$, $b \geq (\Theta(\xi) - \frac{2}{\lambda})n$;

such that

- For all $m \in \{0,1\}^b$, $\Pr[\mathsf{Dec}_{\mathsf{sd}}(AWGN_{\sigma_r^2}(\mathsf{Enc}_{\mathsf{sd}}(m))) = m] \geq 1 - 2^{-\lambda}$;
- For all $m, m' \in \{0,1\}^b$,

$$SD\big((\mathsf{sd}, AWGN_{\sigma_a^2}(\mathsf{Enc}_{\mathsf{sd}}(m))),\ (\mathsf{sd}, AWGN_{\sigma_a^2}(\mathsf{Enc}_{\mathsf{sd}}(m')))\big) \leq 2^{-\lambda}.$$

A.1.3 Using Discrete Gaussian Distributions

Continuous Gaussian distributions cannot be used directly in our blueprint, since they require working with real numbers with infinite precision. We therefore need to “quantize” these numbers in some form. The two natural approaches for doing that are either to round them to some finite precision, or to switch working with discrete Gaussian distributions [22]. Either way, the share sizes will grow linearly with the precision that we use, so it is crucial to analyze the precision needed for error-correction. The effect of rounding on error correction is harder to gauge, especially since the magnitude of the rounding errors grows with $\sqrt{n}$ (where n is the code dimension). Below we therefore sketch an approach that uses discrete Gaussians.

Recall that a discrete Gaussian distribution over a point lattice $\Lambda \subset \mathbb{R}^n$ is a probability distribution over Λ where each point $\vec{x} \in \Lambda$ is assigned probability mass proportional to the Gaussian probability density function. Namely, for a parameter $s \in \mathbb{R}$, denote $\rho_s(\vec{x}) := \exp(-\pi\|\vec{x}/s\|^2)$ and $\rho_s(\Lambda) = \sum_{\vec{x}\in\Lambda} \rho_s(\vec{x})$. Then the discrete Gaussian distribution over Λ with parameter $s \in \mathbb{R}$ (centered at the origin), denoted $D_{\Lambda,s}$, assigns to each $\vec{x} \in \Lambda$ the probability mass $D_{\Lambda,s}(\vec{x}) := \rho_s(\vec{x})/\rho_s(\Lambda)$.

An extensive line of work, starting with Micciancio and Regev [22], established that Discrete Gaussians inherit most of the statistical properties of their continuous counterparts, as long as the parameter s is “sufficiently larger that the precision of Λ”. Specifically, [22] defined the *smoothing parameter* of Λ (relative to some target deviation ϵ), that captures how large the parameter s needs to be for $D_{\Lambda,s}$ to resemble the continuous distribution upto $O(\epsilon)$. Here we only use the fact that for the integer lattice $\mathbb{Z}^n$ and any $\epsilon, \gamma \in \mathbb{R}$, we have $\eta_\epsilon(\gamma \cdot \mathbb{Z}^n) \leq \gamma \cdot \sqrt{\ln(2n(1+1/\epsilon))/\pi}$ (cf. [22, Lemma 3.3]). Specifically, setting $\epsilon = 2^{-\lambda}$ we get

$$\eta_{2^{-\lambda}}(\gamma \cdot \mathbb{Z}^n) \leq \gamma \cdot \sqrt{\frac{\ln(2n(1+2^\lambda))}{\pi}} < \gamma \cdot \sqrt{\ln n + \lambda}.$$

While we could not find in the literature any treatment of error correction as applied to discrete Gaussians, we take the extensive literature on the statistical properties as evidence that the error-correction techniques for continuous Gaussians should still work. Specifically, for discrete Gaussians over $\gamma \cdot \Lambda^n$, the secrecy/correctness error should not increase by more than $O(\epsilon)$, provided that we always use parameter $s \geq \eta_\epsilon(\gamma \cdot \mathbb{Z}^n)$.

A.2 A Discrete AWGN Construction

Instantiating the approach above with precision γ seem to require that *all* the distributions that we use will have parameter of at least the smoothness factor. Since in our construction we give a party with weight w an error component with parameter $s_w \sim \sqrt{w}$, then we need to use a small enough γ so that even for the smallest non-zero weight (which could be $w = 1$) already has a large enough parameter $s_w \geq \eta_{2^{-\lambda - \log N}}(\gamma \cdot \mathbb{Z}^n)$. (The $\log N$ factor comes due to the fact that we have N such distributions, one per party.) That is, we roughly need $\gamma \approx 1/\sqrt{\ln n + \lambda + \log N}$.

On the other hand, for the largest weights and (which could be as large as $\Omega(W)$), and certainly for the public share, we need to use numbers of size at least $\sqrt{W}$. Hence, each entry in our code would require $O(\log(\sqrt{W}/\gamma)) = O(\log(W) + \log\lambda + \log\log n + \log\log N)$ bits to specify. We could use the rounding technique from Figure 2 to remove the dependence on W, replacing the $\log W$ term by $\log(N/\epsilon)$. This means that the number of bits to specify each entry is $O\left(\log N + \log\lambda + \log(1/\epsilon) + \log\log n\right)$. Since we always have $n = \mathsf{poly}(\lambda)$, we can ignore the $\log\log n$ term above.

We now can set $\sigma_r = \sqrt{1-\beta}$ and $\sigma_a = \sqrt{1-\alpha}$, and consider the wiretap channel with receiver channel $D_{\gamma\mathbb{Z}^n,\sigma_r}$ and adversary channel $D_{\gamma\mathbb{Z}^n,\sigma_a}$. Since these distributions are above the smoothing parameter (wrt $\epsilon = 2^{-\lambda/N}$), we can expect the gap between their capacities to be similar to their continuous counterparts, namely we expect $\xi := c(D_{\gamma\mathbb{Z}^n,\sigma_r}) - c(D_{\gamma\mathbb{Z}^n,\sigma_a}) = \Theta(\epsilon)$.

Plugging the parameters from Appendix A.1.2 we would get $n = \max\left(\mathsf{poly}(\frac{1}{\epsilon}), \lambda^2, O(\frac{\lambda}{\epsilon^2})\right)$, so the share size is

$$\max\left(\mathsf{poly}(\frac{1}{\epsilon}), \lambda^2, O(\frac{\lambda}{\epsilon^2})\right) \cdot O\left(\log N + \log\lambda + \log(1/\epsilon)\right).$$

With seed size $k = \Theta(\epsilon)n$ and message size $b \approx (\Theta(\epsilon) - \frac{2}{\lambda})n$. This implies a rate $|\text{secret}|/|\text{share}| = O(\epsilon/(\log N + \log\lambda + \log(1/\epsilon)))$, which is not as good as for the BSC. However, it is plausible that the $\mathsf{poly}(1/\epsilon)$ term for decoding AWGN channels is better than for BSC, in which case the concrete share sizes or short secrets could still be smaller.