On Information-Theoretic Secure Multiparty Computation with Local Repairability

Daniel Escudero[1⋆], Ivan Tjuawinata[2⋆⋆], Chaoping Xing[3⋆⋆⋆]

[1] J.P. Morgan AI Research and J.P. Morgan AlgoCRYPT CoE, New York, U.S.A.
[2] Strategic Centre for Research on Privacy-Preserving Technologies and Systems, Nanyang Technological University, Singapore
[3] Shanghai Jiao Tong University, Shanghai, China

Abstract. In this work we consider the task of designing information-theoretic MPC protocols for which the state of a given party can be recovered from a small amount of parties, a property we refer to as *local repairability*. This is useful when considering MPC over dynamic settings where parties leave and join a computation, a scenario that has gained notable attention in recent literature. Thanks to the results of (Cramer *et al.* EUROCRYPT'00), designing such protocols boils down to constructing a linear secret-sharing scheme (LSSS) with good locality, that is, each share is determined by only a small amount of other shares, that also satisfies the so-called multiplicativity property. Previous constructions that achieve locality (*e.g.* using locally recoverable codes—LRCs) do not enjoy multiplicativity, and LSSS that are multiplicative (*e.g.* Shamir's secret-sharing) do not satisfy locality. Our construction bridges this literature gap by showing the existence of an LSSS that achieves both properties simultaneously.

Our results are obtained by making use of well known connection between error correcting codes and LSSS, in order to adapt the LRC construction by (Tamo & Barg, IEEE Transactions on Information Theory 2014) to turn it into a LSSS. With enough care, such coding-theoretic construction yields our desired locality property, but it falls short at satisfying multiplicativity. In order to address this, we perform an extensive analysis of the privacy properties of our scheme in order to identify parameter regimes where our construction satisfies multiplicativity.

Finally, since our LSSS satisfies locality, every share is determined by a small amount of shares. However, in an MPC context it is not enough to let the (small set of) parties to send their shares to the repaired party, since this may leak more information than the regenerated share. To obtain our final result regarding MPC with local repairability, we construct a lightweight MPC protocol that performs such repairing process without any leakage. We provide both a passively secure construction (for the *plain* multiplicative regime) and an actively secure one (for *strong* multiplicativity).

⋆ daniel.escudero@protonmail.com
⋆⋆ ivan.tjuawinata@ntu.edu.sg
⋆⋆⋆ xingcp@sjtu.edu.cn

1 Introduction

Secure multiparty computation (MPC) is a set of techniques that enables a set of n mutually distrustful parties $P_1, \ldots, P_n$ to securely compute a function on secret data, while revealing only its output, even if some unknown subset t of the parties is corrupted by an adversary. Let us represent the function to be computed securely as an arithmetic circuit comprised of addition and multiplication gates over a finite field $\mathbb{F}$. A popular and successful approach to building MPC protocols consists of letting the parties obtain *secret-shared* versions of the inputs, that is, each party holds a *share* of each input to the computation in such a way that the shares of the corrupted parties collectively leak nothing about the inputs, and yet, certain allowed sets of shares can reconstruct the underlying secret. Then, the parties engage in some interactions in order to securely compute shares of each intermediate wire in the circuit until they reach shares of the output, at which point they use the reconstruction procedure to learn the result. Many notable protocols follow this paradigm, both in the honest majority setting [32,33,20,9,34] (where the adversary corrupts at most a minority of the parties) and also with dishonest majority [6,24,23,43,44,4] (where, in contrast, the adversary may control all but one of the parties). Furthermore, MPC protocols are also categorized by the level of security they achieve (e.g. computational or information-theoretic), or by the type of adversary they tolerate (e.g. passive/semi-honest or active/malicious).

In this work we focus specifically in the context of passive security with $t < n/2$ and active security with $t < n/3$, both with perfect security. In this case, a common template is to use Shamir's secret-sharing [8,65], where the shares of a secret $s \in \mathbb{F}$ are given by $(f(1), \ldots, f(n))$, where $f(\mathtt{X})$ is a random polynomial over $\mathbb{F}$ of degree $\leq t$, subject to $f(0) = s$. This scheme satisfies t-privacy (meaning any t shares do not leak any information about the secret, which ensures the adversary does not learn any sensitive information) and $(t+1)$-reconstruction (meaning that any $t+1$ shares together can reconstruct the secret). Furthermore, this scheme is *linear*, which means that the parties can locally add their shares of two secrets to obtain shares of the sum; this property enables the parties to process addition gates non-interactively. Finally, another important property is that the parties can locally multiply their shares of two secrets to obtain degree-$2t$ sharings of their product; in particular, if $t < n/2$, so $2t < n$, all the parties together can still reconstruct the product of the secrets, and if $t < n/3$ the $n - t$ honest parties on their own can do this, without the help of the t corrupt parties. These properties are accordingly called *multiplicativity* and *strong multiplicativity*, and they are key in obtaining MPC for $t < n/2$ with passive security and $t < n/3$ with active security, respectively, and in fact there is a long series of works that relies specifically on this construction (*e.g.* [5,16], to cite a few). Even more, it was shown in [18] that MPC is in general possible from *any* linear secret-sharing scheme (LSSS) that satisfies multiplicativity (for $t < n/2$ with passive security) and strong multiplicativity (for $t < n/3$ with active security) and in fact, it is currently unknown whether we can obtain this type of MPC without using these properties.

MPC with repairing parties. In protocols that follow the secret-sharing paradigm, the "state" of a party during a protocol execution is typically given by the set of shares it holds. Depending on the secret-sharing scheme used and the adversarial setting, this "state" is usually determined by the shares held by the other parties. For example, in Shamir's secret-sharing any $t + 1$ sharings together determine the polynomial $f(\mathtt{X})$ and hence determine the secret $s = f(0)$, but even more interestingly, they also determine *any other share* $f(i)$. This way, if a party P_i needs to learn its "state" $f(i)$, the other parties can engage in a lightweight MPC protocol where each party P_j for $j \neq i$ inputs its share $f(j)$, and P_i learns precisely $f(i)$. We will refer to the task of reconstructing P_i's share as *repairing* or *regenerating* this share.

The ability to restore a party's state from the other parties' information is useful once we factor in the fact that, depending on the setting and the function being computed, the execution of an MPC protocol can take a considerable amount of resources and time. During such period of time it is not unreasonable for a party to have the need to learn its state: perhaps the party crashed and lost its state, or it had network issues and got disconnected, or possibly a new party is joining the computation (which is useful for example to add diversity to the computation, preventing collusions). However, the simple idea sketched above suffers from a massive drawback: the party joining must receive messages from *all* of the other computing parties in order to obtain its share. This is a huge blocker, especially in large-scale scenarios. Compare this to, for example, other large-scale distributed scenarios such as permissionless blockchains: consensus is maintained by letting multiple parties hold the same view of the underlying ledger, and whenever a new participant wishes to join the network, he or she only needs to contact a small subset of nodes in order to receive the current state, at which point the new node can become an active member. In fact, one can argue that part of the scalability of such systems comes from the fact that their underlying networks are comprised of "local" committees, where each party connects to a subset of the nodes, and messages are propagated via network flooding and echoing.

The above discussion sets the stage for the following question:

> *Is it possible to design MPC protocols with repairing ability, in such a way that regenerating the state of a party does not require communication from all of the other computing parties?*

1.1 Our Contribution

Our work makes substantial progress in addressing the question above by introducing a linear secret-sharing scheme that simultaneously (1) allows for efficient share repairing and (2) is suitable for the design of honest majority MPC protocols. Shamir's secret-sharing as described previously is a good example of a scheme that satisfies (2)—and in fact is one of the most widely used building blocks in honest majority MPC—but it does not satisfy (1) as repairing requires communication from a large amount of parties. In contrast, the literature of

secure distributed storage (see for example [28,45,70]), where multiple parties hold shares of a secret but only for the purpose of storing it (in contrast to *computing* on it as in MPC), already considers secret-sharing schemes that satisfy (1). Efficient regeneration of shares is crucial in this context in order to preserve data availability and to achieve this, multiple works make use of *locally recoverable codes* (LRC), which enable shares to be regenerated only from a small subset of shares. Unfortunately, such works did not consider multiplicativity—needed for item (2)—when using the efficient LRC constructions from [66,69,53,52].

We reconcile the state of affairs from above by considering a linear secret sharing scheme that satisfies the two items required, which makes it suitable in the context of MPC where parties leave (*e.g.* due to crashing) and rejoin a given computation. We achieve this by proposing an efficient and secure share repairing lightweight protocol, and we also investigate the multiplicativity and strong multiplicativity properties of our lsss, for certain corruption regimes. Our scheme, unlike Shamir's secret-sharing, is not a *threshold* scheme, meaning that while its privacy threshold is t, its reconstruction threshold r is strictly larger than $t+1$. Such schemes are called *ramp*. Given the context above, we state the following result pertaining the properties of our LSSS:

Corollary 1 (Simplification of Theorems 1 and 2) *For any prime power q and positive integers v, n such that $v \mid n$, there exists a secure repairable linear ramp secret-sharing Σ for n players over $\mathbb{F}_q$ with reconstruction threshold r and privacy threshold t, where the gap is $r - t = O\left(\frac{n}{v}\right)$, and such that any share can be recovered by contacting v other players. Furthermore*

- *There exists a family of instantiations of Σ with $t < r \leq \frac{n}{2}$ that is multiplicative where $t = n\left(\frac{1}{2} - O\left(\frac{1}{v}\right)\right) + O(1)$.*
- *There exists a family of instantiations of Σ with $t < r \leq \frac{n}{3}$ that is strongly multiplicative where $t = n\left(\frac{1}{3} - O\left(\frac{1}{v}\right)\right) + O(1)$.*

By setting $v \approx n$ we obtain parameters comparable to Shamir's: multiplicativity for $t \approx n/2$ and strong multiplicativity for $t \approx n/3$, but we do not save in terms of repairing since essentially all shares are needed to regenerate an additional share. Interesting regimes occur when we take $v \ll n$. For example, by taking $v \approx n/c$ we obtain multiplicativity with $t \approx n/2 - c$, and strong multiplicativity with $t \approx n/3 - c$. This shows we can save a *multiplicative* factor of c in terms of locality, while only sacrificing the corruption tolerance by an *additive* term of c. We remark that our actual construction involves several parameters, and Corollary 1 is a concrete instantiation of the family of LSSSs we construct. For details, we refer the reader to Section 3 where we present our construction and present its parameters more thoroughly.

On share privacy and static adversaries. In our LSSS, each share is determined by at most v other shares, where v can be chosen to be much smaller than n. More precisely, the set of n parties will be partitioned in a series of groups of size $v+1$, where v shares within a group can be used to reconstruct the remaining share of the group. However, this means that if an adversary corrupts v parties

of a single group, then via locality the adversary can learn the share of the remaining uncorrupted party. This may not affect privacy of the secret since even with these extra shares the adversary may not have enough information to reconstruct the secret, but still violates the privacy of the honest party, whose share has been revealed. We consider the notion of *share privacy*, where the share of each honest party must also be kept secret from the adversary, and similar to committee-based approaches in MPC (e.g. [22]), we achieve it by assuming that the adversary is *static*, meaning that he chooses which parties to corrupt before the actual protocol execution starts, and that the assignment of parties into groups is done uniformly at random. From this, via a careful choice of parameters and a non-trivial analysis of probabilities, we are able to show our repairing protocol and hence the whole secret sharing scheme is statistically secure, when the group size (and hence the number of parties) is sufficiently large. We remark that the restriction of the adversary being static is required for any repairable secret sharing scheme that provides share privacy. In fact, the same assumption of a static adversary can also be found in other works on repairable secret sharing (see, for example [46,1,42,60,48]).

Secure protocols for regenerating shares. In the context of repairing a given share, sending the v needed shares for regeneration *in the clear* may leak more than the intended share to regenerate, and in fact it may violate share privacy as considered above. To address this, instead of the v parties sending their shares directly to the party being repaired, these parties run a lightweight MPC protocol in which the v parties input their shares and the receiving party receives *only* its own regenerated share. We refer to such procedure as a *repairing protocol with locality* v. In this work we also present explicit and efficient repairing protocols for our secret-sharing scheme, which enables a party to learn its share while communicating with only v parties, and without leaking anything beyond this share. As above, our repairing protocols only tolerate *static adversaries*, where the set of corrupt parties is fixed before any protocol execution.

MPC results. Coupling the properties of our LSSS together with our repairing protocols and the results in [18], we obtain the following as corollaries:

Corollary 2 (MPC with efficient repairing) *Let q be a prime power, and let v, n be positive integers such that $v \mid n$ and $v = \ln^{(1+\varepsilon)} n$ for some $\varepsilon > 0$. Assume that $n = \Omega(\kappa)$, where κ is the statistical security parameter. Then there exist statistically secure MPC protocols for general arithmetic circuits, protecting against a static adversary corrupting t parties, and having repairing protocols with locality v, with either one of the following properties:*

- *Passive security with $t = n\left(\frac{1}{2} - O\left(\frac{1}{v}\right)\right) + O(1)$.*
- *Active security with $t = n\left(\frac{1}{3} - O\left(\frac{1}{v}\right)\right) + O(1)$.*

1.2 Related Work

Repairable secret-sharing. The notion of locally repairable codes, which implicitly lies at the core of our construction, is used in a wide variety of scenarios such as

distributed storage systems, or DSS, for short. In this context, a piece of data is encoded and stored in several nodes with the goal of ensuring its availability even if some nodes fail/crash, and at the same time, possibly, providing some notion of privacy. Among the desired properties for such system we find repairability (see, for example, [14,15,49,73]), which, intuitively, allows any entry of a given codeword to be determined by partial information obtained from some of the other entries of the codeword. When translating these notions to the secret-sharing setting, we arrive at the concept of repairable secret sharing.

The study of secure repairable secret sharing was firstly proposed by Herzberg *et al.* [39].[4] In that work, the authors proposed a reparing mechanism to enroll new parties based on Shamir's secret sharing. Since then, there have been studies that follow a similar direction with the help of error correcting codes, publicly-verifiable secret sharing schemes, bivariate polynomial secret sharing scheme and vector space secret sharing schemes (see, for example, [71,74,62]). A survey on the study of repairable secret sharing schemes can be found in [46].

Repairable error-correcting codes. The concept of repairability has been well-studied in the field of error correcting codes, in particular in its relation with distributed storage systems. Regenerating codes constitute a family of error correcting codes that was proposed by Dimakis *et al.* [26] where, given any codeword of the code, any of its entries can be recovered from some partial information from some of the other entries. Such property enables regenerating codes to be used to encode data in several nodes where failure in a node can be repaired by downloading some information from some other nodes. A bit more precisely, we may encode a piece of data $\mathfrak{D}$ in N nodes where each node stores δ bits of data in such a way that $\mathfrak{D}$ can be recovered by downloading the information stored in any $K \leq N$ nodes, with K ideally being much smaller than N. The regenerating capability ensures that if a node fails, it can contact $K \leq D < N$ nodes and download κ bits from each of the contacted node to regenerate the data to be stored in the failed node. Here the *repair bandwidth* is the total amount of bits needed to perform a repair process, which is $D \cdot \kappa$ in the case above.

Some families of regenerating codes that are constructed with the objective of minimizing either storage or bandwidth are called Minimum Storage Regenerating (MSR for short) codes and Minimum Bandwidth Regenerating (MBR for short) codes, respectively. There have been numerous studies investigating both MSR and MBR codes. It was shown that we can construct codes from both families using product-matrix constructions [57]. There have also been some studies on regenerating codes in various other directions such as impossibility results (e.g. [64]), existential bounds (e.g. [10,68,3,61]) and explicit constructions (e.g. [31,58,72]).

[4] More precisely, [39] studies *proactive secret-sharing schemes* in which shares of a given secret must be "refreshed". However, in Section 4 of their paper, the authors argue that share recovery is essential to have a secure proactive SS. In addition, in several other references on repairable secret-sharing schemes, [39] is credited to be the first to propose the concept of repairing shares (which is either corrupt or lost).

Note that in practice, the nodes may be spread around the world, and in such a case the distance between nodes may become very large. Hence, instead of just focusing on the repair bandwidth, we may also be interested in keeping the amount of contacted nodes small. This is captured by the metric of *locality*. The study of this concept, together with the construction of codes with small locality, were first pioneered in [30,38,40]. There have then been many studies on both bounds and constructions of locally repairable codes; see for example [69,41,50,37,11,51].

Securely repairing shares in repairable secret-sharing. Despite the numerous studies on the concept of repairable error correcting codes, such studies only focus on repairing failing nodes without considering the privacy of either the original data being encoded, or the data stored in different nodes. Furthermore, since these studies are typically set in the domain of distributed storage systems, they consider *static* data that is not manipulated to perform arbitrary computations,[5] which is the case in the field of secure MPC.

Due to the close relation between error correcting codes and secret sharing schemes, the concept of repairability has also been considered in the latter. To enable this, some notion of *security* has been added to the concept of regenerating codes, (e.g. [1,42,60]) This study was first initialized by Pawar *et al.* [54]. Secure repairable secret sharing schemes have subsequently been proposed based on various techniques such as enrollment protocols [35,67], combinatorial design [67], linearized polynomials [63] and regenerating codes [63,57,59]. However, in all these studies, the adversary model that is considered is that of a passive adversary who can learn the data stored in a set of players and the messages received by a different set of players, without the ability to arbitrarily modify the behaviour of those players.

To the best of our knowledge, there has only been one work on the construction of regenerating codes having security against somewhat "active" adversaries [48]. In this work, Li *et al.* constructed a repairable secret sharing scheme by first masking the message with the output stream of a linear feedback shift register (LFSR) before encoding the masked message with an MSR encoding. In their work, they show that the resulting regenerating code is secure against a passive eavesdropper, but enhanced so that any malicious modification of any message can be detected and corrected with probability of at least $1 - \frac{1}{q}$, where q is the order of the underlying finite field. However, in that work, the privacy of the shares is not considered during the repairing process, while the privacy of the secret is obtained via the security guarantee of the underlying block cipher. Furthermore, the authors do not consider the multiplicative property of the scheme, which is an essential property for the application of secret sharing scheme in the design of actively secure MPC schemes. Such property is most likely not satisfied due to the encryption step before the share generation. In other words, to the best of our

[5] If the encoding process is linear then computation represented by simple linear operations is possible. However, the calculation of the product of two encoded values is not easy to achieve, which is where the concepts of multiplicativity and strong multiplicativity become useful.

knowledge, our construction from Section 5 constitutes the first repairable secret sharing scheme suitable for secure multiparty computation, satisfying security against an active adversary.

Enabling parties to join a secure computation. The problem of handling involuntary crashes in MPC has received quite some attention recently, and several works are dedicated to the study of such protocols [27,2,36,17,29,21]. However, among those, to the best of our knowledge, only [17,29,21] enable parties to rejoin the computation, and moreover, only [21] allows parties to rejoin while possibly missing all the intermediate messages sent to them while being offline. Our work enables parties to join a computation not only if they miss messages while being offline, but even if they were never a previous participant in the protocol to begin with. However, we do not achieve optimal privacy and reconstruction parameters (in fact, our scheme is a ramp scheme, which has a gap between the privacy and reconstruction thresholds), while the works mentioned above show and achieve lower bounds. Furthermore, these works build on top of "standard" Shamir's secret-sharing and hence require communication from essentially all the parties, while our work considers a substantial modification to this scheme that enables local repairability.

Remark 1 (On "YOSO-fying" or "Fluidifying" our protocols). Interestingly, both YOSO and Fluid MPC [17,29] and their follow-ups [7,55,25,12] consider MPC in a setting where the set of parties change dynamically in every round. Furthermore, YOSO focuses on removing the static-corruption assumption from committee-based MPC protocols, and an interesting research direction involves using YOSO-like protocols to remove the static-adversary assumption from our work. Another interesting direction consists of using our secret-sharing scheme to improve the efficiency of YOSO/Fluid protocols, in particular, reducing the communication overhead of the re-sharing step (whereby a committee passes a shared secret to the next committee), which is typically the bottleneck in these protocols. Our scheme allows for each share to be determined from a small amount of shares, which may indeed improve efficiency. We leave this for future work.

1.3 Organization

This paper is organized as follows. In Section 2, we briefly define some basic notations and discuss some basic concepts that will be useful in our discussion. Section 3 contains our main LSSS Σ as Algorithm 1, together with the analysis of its main properties. Next, we consider how to securely repair shares for the multiplicative variant of our construction with passive security in Section 4. This is then extended to active adversaries for the strongly multiplicative variant of our scheme in Section 5. Finally, we provide some discussion on related work in Section 1.2.

q	Prime power, number of field elements
$\mathbb{F}_q$	The finite field with q elements
n	Number of players
τ	Number of corrupted players
P_i	The i-th player, $1 \leq i \leq n$
v	Number of involved parties in secure repairing scheme (size of each local group: $v+1$, which divides $q+1$)
H	Multiplicative subgroup of $\mathbb{F}_q^*$ of size $v+1$
m	Number of cosets of H considered, or number of local groups, where $m \leq \frac{q-1}{v+1}, n = m(v+1)$
$\beta_1, \ldots, \beta_m$	Coset leaders of H
$\ell(x)$	$\ell(x) = \prod_{\alpha \in H}(x - \alpha)$
ρ	$\rho \in \mathbb{F}_q \setminus \ell(\mathbb{F}_q)$
$g(x)$	$g(x) = \ell(x) + \rho$
$\gamma_0, \gamma_1, \ldots, \gamma_n$	Evaluation points belonging to $\{0\} \cup \left(\bigcup_{i=1}^{m} \beta_i H\right)$, where $\gamma_0 = 0$.
$f(x)$	Polynomial used to share $s, s = \sum_{j=0}^{w} a_{0j} g(0)^j$. It holds that $f(x) = \sum_{i=0}^{d-1} \left(\sum_{j=0}^{w} a_{i,j} g(x)^j\right) x^i \in \mathbb{F}_q[x]$
d	Degree of $f(x)$ (plus one) when $g(x)$ is fixed to be a constant, upper bound on corrupted parties in each local group, $d \leq v$
w	$w+1$ is the minimum number of local groups required to recover the secret, $w < m$
$s \in \mathbb{F}_q$	Secret
t	Privacy level
r	Reconstruction level
$h(x)$	Polynomial mask for repairing process
$h_i(x)$	Share of polynomial mask generated by P_i

Table 1. Table of Notations

2 Preliminaries

In this section, we briefly discuss some notations that are used throughout the manuscript. As a useful reference, a complete table of notations can be found in Table 1

For a prime power q, we denote by $\mathbb{F}_q$ the finite field with q elements. We also define $\mathbb{F}_q[\mathtt{X}]$ to be the set of polynomials over $\mathbb{F}_q$. For any positive integer N, define $\mathbb{F}_q^N$ the set of vectors of length N over $\mathbb{F}_q$. Also, for any positive integer n, we denote by $[n] = \{1, \ldots, n\}$. Let $v \geq 2$ be a positive integer such that $(v+1)|(q-1)$. This implies that there exists a unique multiplicative subgroup H of $\mathbb{F}_q^*$ of size $v+1$. Observe that H partitions $\mathbb{F}_q^*$ to $\frac{q-1}{v+1}$ disjoint cosets. Let such cosets be $\beta_1 H, \beta_2 H, \ldots, \beta_{\frac{q-1}{v+1}} H$.

In the following, we define a function that is a constant in each coset, which will be used as an ingredient in our secret sharing construction.

Lemma 3. *Let* $\ell(\mathtt{X}) = \prod_{\alpha \in H}(\mathtt{X} - \alpha)$. *Then* $\ell(\mathtt{X})$ *is a constant in each coset and* $\ell(0) \neq 0$. *Furthermore, for any* $\beta, \gamma \in \mathbb{F}_q, \ell(\beta) = \ell(\gamma)$ *if and only if either* β, γ *belong to the same coset or* $\beta = \gamma = 0$.

Proof. Indeed, for any $\beta_i \alpha' \in \beta_i H$ with $\alpha' \in H$, we have

$$\ell(\beta_i \alpha') = \prod_{\alpha \in H} (\beta_i \alpha' - \alpha) = (\alpha')^{v+1} \prod_{\alpha \in H} (\beta_i - \alpha(\alpha')^{-1}) = \prod_{\alpha \in H} (\beta_i - \alpha) = \ell(\beta_i).$$

This proves the first part.

Now suppose that $\ell(\beta) = \ell(\gamma)$ and at least one of β, γ is a nonzero element. Then, from the first statement we already proved, $\ell(\lambda) - \ell(\beta) = 0$ for all $\lambda \in \beta H \cup \gamma H$. As $\deg(\ell(\mathtt{X}) - \ell(\beta)) = |H|$, we must have $|\beta H \cup \gamma H| \leqslant |H|$. This implies that both β, γ are nonzero and they belong to the same coset. □

2.1 Secret Sharing Schemes

Let $\mathtt{P} = \{P_1, \ldots, P_n\}$ be a finite set of players. A *forbidden set* $\mathcal{F}$ is a family of subsets of $\mathtt{P}$ such that for any $A \in \mathcal{F}$ and $A' \subseteq A$, we must have $A' \in \mathcal{F}$. For any $t < n$, we define $\mathcal{F}_{t,n}$ to be the forbidden set containing all subsets of $\mathtt{P}$ of size at most t. On the other hand, a *qualified set* Γ is a family of subsets of $\mathtt{P}$ such that for any $B \in \Gamma$ and $B \subseteq B'$, we must have $B' \in \Gamma$. For any $r \leq n$, we define $\Gamma_{r,n}$ to be the qualified set containing all subsets of $\mathtt{P}$ of size at least r. For any forbidden set $\mathcal{F}$ and a qualified set Γ over $\mathtt{P}$ such that $\mathcal{F} \cap \Gamma = \emptyset$, the pair $(\mathcal{F}, \Gamma)$ is called an *access structure*.

A secret sharing scheme with access structure $(\mathcal{F}, \Gamma)$ over $\mathbb{F}_q$ on $\mathtt{P}$ is a pair of functions $(\mathsf{Share}, \mathsf{Rec})$ where Share is a probabilistic function that calculates the random shares for the n players given the secret. For any secret $S \in \mathbb{F}_q$, if $(\mathbf{S}_1, \ldots, \mathbf{S}_n) = \mathsf{Share}(S)$, for any $A \subseteq \mathtt{P}$, we denote by $\mathbf{S}^A = (\mathbf{S}_i)_{P_i \in A}$, the vector containing the shares of all players $P_i \in A$. On the other hand, Rec accepts shares from a set of players in $\mathtt{P}$ and attempt to recover the original secret that satisfies the following requirements:

1. For any $B \in \Gamma$, given the shares of all players $P_i \in B$, Rec returns the original secret S.
2. For any $A \in \mathcal{F}$, the shares of $P_i \in A$ does not give any information regarding the secret. That is, the distribution of the shares $\mathbf{S}^A$ of players in A is independent of the original secret S.

A secret sharing scheme with access structure $(\mathcal{F}, \Gamma)$ such that $\mathcal{F}_{t,n} \subseteq \mathcal{F}$ and $\Gamma_{r,n} \subseteq \Gamma$ for some $0 < t < r < n$ is called a *ramp secret sharing scheme providing t-privacy and r-reconstruction*. If $r = t + 1$, we call it a threshold secret sharing scheme.

Next we discuss the notions of linearity, multiplicativity and strong multiplicativity. We follow the definitions given in [19].

A *linear secret sharing scheme* over $\mathbb{F}_q$ on n players with secret space and share space $\mathbb{F}_q$ is a pair $(\mathsf{Share} : \mathbb{F}_q \to \mathbb{F}_q^n,\ \mathsf{Rec} : \mathbb{F}_q^* \to \mathbb{F}_q \cup \{\perp\})$ such that the set $\{(s, \mathsf{Share}(s)) : s \in \mathbb{F}_q\}$ is an $\mathbb{F}_q$-subspace of $\mathbb{F}_q \times \mathbb{F}_q^n$ and for any $s \in \mathbb{F}_q, \mathsf{Rec}(\mathsf{Share}(s)) = s$. A linear secret sharing scheme Σ is said to be multiplicative if there is a vector $\mathbf{r} \in \mathbb{F}_q^n$ such that for any two secrets $s, s' \in \mathbb{F}_q$ with their respective shares $\mathsf{Share}(s) = (s_1, \ldots, s_n)$ and $\mathsf{Share}(s') = (s'_1, \ldots, s'_n)$ where $s_i, s'_i \in \mathbb{F}_q$ for any $i = 1, \ldots, n$, $\mathbf{r} \circ (s_1 s'_1, \ldots, s_n s'_n) = s \cdot s'$ where $\circ$ represents the standard inner product.

Lastly, a linear secret sharing Σ is said to be t-strongly multiplicative if it has t-privacy and for any two secrets $s, s' \in \mathbb{F}_q$ such that $(s_1, \ldots, s_n) = \mathsf{Share}(s), (s'_1, \ldots, s'_n) = \mathsf{Share}(s'), ss'$ can be recovered from any $n - t$ entries of $(s_1 s'_1, \ldots, s_n s'_n)$.

A well-known example of a linear threshold secret sharing scheme is Shamir's secret sharing scheme. For two positive integers t and n such that $t < n < q$, a (t, n)-Shamir's secret sharing scheme over $\mathbb{F}_q$ is a linear secret sharing scheme for n parties with providing t-privacy and $t + 1$ reconstruction with both secrets and shares being elements of $\mathbb{F}_q$. The share generation is done in the following way. First, we choose n non-zero pairwise distinct elements of $\mathbb{F}_q$, say $\alpha_1, \ldots, \alpha_n$, and assign each element to different players. Given the secret $s \in \mathbb{F}_q$, first, we randomly choose a polynomial $f(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]$ of degree t such that $f(0) = s$. Then the share for the player assigned to the element α_i is defined to be $f(\alpha_i) \in \mathbb{F}_q$. It can be shown that this secret sharing scheme provides t-privacy and $t + 1$ reconstruction. Furthermore, such secret sharing scheme is linear, multiplicative when $t < \frac{n}{2}$ and strongly multiplicative when $t < \frac{n}{3}$.

2.2 Linear Codes

In this section, we briefly discuss the concept of linear codes.

Definition 1 (Linear Codes) *Let n, k, d be non-negative integers such that d and k are at most n. A linear code C over $\mathbb{F}_q$ with parameter $[n, k, d]$ is a subspace $C \subseteq \mathbb{F}_q^n$ of dimension k such that for any non-zero $\mathbf{c} \in C \setminus \{\mathbf{0}\}, wt_H(\mathbf{c}) \geq d$ where for any vector $\mathbf{x} = (x_1, \ldots, x_n), wt_H(\mathbf{x})$ is defined to be the Hamming weight of $\mathbf{x}$, i.e., $wt_H(\mathbf{x}) = |\{i : x_i \neq 0\}|$. It is well known that a code of minimum distance d can uniquely correct any error of Hamming weight $\left\lfloor \frac{d-1}{2} \right\rfloor$.*

Next, we recall the definition of Reed-Solomon codes.

Definition 2 *Let q be a prime power and let n and k be positive integers such that $k \leq n \leq q$. Fix $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ where α_i is pairwise distinct. We define $\mathbb{F}_q[\mathtt{X}]_{<k}$ to be the set of polynomials over $\mathbb{F}_q$ of degree at most $k - 1$. For any $f(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]$, define $\mathbf{c}_{f,\boldsymbol{\alpha}} = (f(\alpha_1), \ldots, f(\alpha_n)) \in \mathbb{F}_q^n$. The Reed-Solomon code $RS_{n,k,\boldsymbol{\alpha}}$ is defined to be the set of vectors $\mathbf{c}_{f,\boldsymbol{\alpha}}$ with $f(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]_{<k}$, that is,*

$$\mathsf{RS}_{n,k,\boldsymbol{\alpha}} = \{(f(\alpha_1), \ldots, f(\alpha_n)) : f(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]_{<k}\}.$$

It is well known that for any choice of $\boldsymbol{\alpha}$, the minimum Hamming distance of $RS_{n,k,\boldsymbol{\alpha}}$ is $n - k + 1$. Hence, it can uniquely correct up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ Hamming errors.

2.3 Security Model

In this work, we consider security against a computationally-unbounded adversary controlling τ out of the n players. In other words, throughout this work, we consider constructions in information-theoretical security setting. Here we consider two types of adversaries depending on the extent of control it has on the players. Firstly, we focus on a semi-honest adversary which learns all the values stored by the players it controls. Secondly, we also consider a malicious adversary which means that the corrupted players do not have to follow the protocol. In either case we assume the corruption is static, i.e. the adversary chooses the parties to corrupt at the beginning of the protocol, and no more parties are corrupted once the execution of the protocol starts.

3 Our Linear Secret-Sharing Scheme with Good Locality

Let q be a prime power, and let $v \geq 2$ be a positive integer such that $(v+1) \mid (q-1)$. Let $n = (v+1)m$ for some integer $m \in \left\{1, \ldots, \frac{q-1}{v+1}\right\}$, and let $r = w(v+1) + d$ for some chosen positive integer $w \leq m-1$ and $d \leq v$. In this section, we discuss the construction of a repairable secret sharing scheme with n players and reconstruction level r. The scheme has t-privacy, where t depends on the value of w: if $w = m-1$ then $t = v(w+1) - 1 = vm - 1$, and if $w < m-1$ then $t = (v-1)(w+1)$.

Let H be the unique multiplicative subgroup H of $\mathbb{F}_q^*$ of size $v+1$. Observe that H partitions $\mathbb{F}_q^*$ to $m = \frac{q-1}{v+1}$ disjoint cosets $\beta_1 H, \beta_2 H, \ldots, \beta_m H$. It can be proven (see Lemma 3 in Section 2) that if $\ell(\mathtt{X}) = \prod_{\alpha \in H}(\mathtt{X} - \alpha)$, then for any $\beta, \gamma \in \mathbb{F}_q$, $\ell(\beta) = \ell(\gamma)$ if and only if either β, γ belong to the same coset or $\beta = \gamma = 0$; furthermore, $\ell(0) \neq 0$. Choose m distinct cosets of H and let $\beta_1, \ldots, \beta_m$ be their coset representatives. We denote the set $\{0\} \cup (\bigcup_{i=1}^{m} \beta_i H)$ by $\{\gamma_0 = 0, \gamma_1, \ldots, \gamma_n\}$. Let $\ell(\mathtt{X})$ be the polynomial defined in Lemma 3. Note that $\ell(\mathbb{F}_q)$ is a subset of $\mathbb{F}_q$ of size $1 + \frac{q-1}{v+1} < q$. Hence there exists $\rho \in \mathbb{F}_q$ such that $-\rho \notin \ell(\mathbb{F}_q)$. Define $g(\mathtt{X}) = \ell(\mathtt{X}) + \rho$. Then $0 \notin g(\mathbb{F}_q)$, and $g(\beta) = g(\gamma)$ if and only if either they belong to the same coset or $\beta = \gamma = 0$. In our LSSS construction each party P_i gets assigned a random element $\gamma_{\pi(i)}$, for some random permutation $\pi : [n] \to [n]$ that is sampled before the adversary sets τ of the parties to be corrupt (recall we only consider *static* adversaries). Note that such random assignment is only done once before the sharing of the first secret and the same assignment will be used for all the subsequent sharings. In an MPC setting, such randomization can be done in the clear by the n parties as a part of the initialization phase; for example, the parties can use a publicly available randomness beacon (see for example [13,47,56]) to generate a random assignment of the parties to their IDs. For notational simplicity we will assume π is the identity function, that is, each P_i gets assigned to γ_i.

Now we are ready to discuss our construction, which can be found in Construction 1. For simplicity in the description we fix the parities (odd/even) of some of the parameters involved. This can be easily generalized to remove such restriction.

At a high level, our LSSS follows a re-interpretation of the local-repairable codes proposed by Tamo and Barg [69], in the context of secret-sharing. As we have already pointed out, the authors in [69] are not concerned with privacy, nor computation, and hence concepts such as t-privacy, multiplicativity or strong multiplicativity is not within the scope of their work. We further note that although the construction of local-repairable codes allows codeword entries to be repaired by the use of small numbers of other entries, such repairing process is designed without privacy as a factor. As it turns out, it is highly non-trivial to perform such analysis, and we do this in Sections 3.1,3.2, and 4 respectively.

Construction 1 *Let q and v be chosen such that $3 \mid (v+1)$ and v is odd. Let n and m be as defined above, with n odd. We also let d be a positive integer such that $d \leq v$. Lastly, we let $w \leq m-1, r = w(v+1)+d$ and we also fix the m distinct cosets of H as well as their coset leaders $\beta_1, \ldots, \beta_m$. The construction is presented as Algorithm 1.*

Algorithm 1 Repairable Secret Sharing Scheme

Require: $S \in \mathbb{F}_q$: the secret to be secretly shared;

1: Randomly select $a_{i,j} \in \mathbb{F}_q$ for $i = 0, 1, \ldots, w$ and $j = 0, 1, \ldots, d-1$ subject to $S = \sum_{j=0}^{w} a_{0j} g(0)^j$;

2: Define $f(\mathtt{X}) = \sum_{i=0}^{d-1} \left(\sum_{j=0}^{w} a_{i,j} g(\mathtt{X})^j \right) \mathtt{X}^i$;

3: **Secret and shares:** Calculate and distribute the shares to the n players where the share for the player assigned γ_i is $S_i = f(\gamma_i)$ for $i = 1, \ldots, n$;

Remark 2 (On two-level Shamir's secret-sharing). A related and simpler construction is to split the parties into groups, use Shamir's secret-sharing to obtain one sharing per group, and secret-share each of these shares, again using Shamir's secret-sharing, among the members of the corresponding group. This scheme has good locality since each "two-level share" is determined by the shares of the parties in the given group. However, it turns out that our scheme can be regarded a refined version of this simple and naive construction, and by our more elaborate analysis we are able to obtain much better parameters. We expand on this discussion in Section A in the Appendix.

3.1 Reconstruction, Multiplicativity and Strong Multiplicativity

Now we consider the reconstruction, multiplicative and strong multiplicative properties of our LSSS. We analyze privacy in Section 3.2. The repairing process for different adversary settings are analyzed in Section 4.

Theorem 1. *Let $\mathbb{F}_q$ be a finite field of q elements, v, m, w, d be positive integers such that $v+1$ divides $q-1, m \leq \frac{q-1}{v+1}, w \leq m-1$ and $d \leq v$. Then the secret sharing scheme Σ for $n = (v+1)m$ players over $\mathbb{F}_q$ constructed in Construction 1 using the parameters q, v, n, m, d, w has the following properties:*

(i) **Reconstruction**: *It has r-reconstruction with $r = w(v+1)+d$,*
(ii) **Multiplicativity**: *The product of two secrets can be recovered as a linear combination of the product of the corresponding shares if $2w(v+1)+2d-1 \leq n$*
(iii) **Strong Multiplicativity**: *Σ is t'-strongly multiplicative if and only if Σ is t'-private and $t' \leq n-(2w(v+1)+2d-1)$.*

Proof. (i) As the degree of g is v, the total degree of $f(\mathtt{X})$ is $w(v+1)+d-1$. Hence, the secret can be reconstructed by any $w(v+1)+d$ shares.

(ii) The product of the corresponding shares forms a Reed-Solomon code with length n and dimension $2(w(v+1)+d-1)+1 = 2w(v+1)+2d-1$. It is only well defined if $2w(v+1)+2d-1 \leq n$. Hence Σ is multiplicative if and only if $2w(v+1)+2d-1 \leq n$.

(iii) By definition, Σ is t' strongly multiplicative if and only if Σ is t'-private and for any two secrets $s, s' \in \mathbb{F}_q$ such that $(s_1, \ldots, s_n) = \mathsf{Share}(s) = (s_1, \ldots, s_n)$ and $(s'_1, \ldots, s'_n) = \mathsf{Share}(s')$, ss' can be recovered from any $n-t'$ entries of $(s_1 s'_1, \ldots, s_n s'_n)$. As discussed above, $(s_1 s'_1, \ldots, s_n s'_n)$ is a codeword of a Reed-Solomon code with length n and dimension $2w(v+1)+2d-1$. Hence it has $2w(v+1)+2d-1$-reconstruction. So Σ is t'-strongly multiplicative if and only if Σ is t'-private and $t' \leq n-2w(v+1)-2d+1$, concluding the proof. □

Parameters for Multiplicativity. To understand better what parameter regimes are attainable with our construction, we discuss some concrete parameter choices that lead to our scheme being multiplicative. We also discuss below the strongly multiplicative case. From Theorem 1, for Σ to be multiplicative, we require $2w(v+1)+2d-1 \leq n$. Since $d \geq 1$, we have $2w(v+1)+2d-1 > 2w(v+1)$. Hence, using $n = m(v+1)$, we must have $w \leq \frac{m}{2} < m$, which implies that (a lower bound on) the privacy threshold is $t = (d-1)(w+1)$. For any choice of $\delta > 0$, assuming that $v+1 \geq \frac{3}{2\delta}$, we may set $d \approx (1-\delta)(v+1)+\frac{1}{2} \leq v$ and $w \approx \frac{m}{2}-(1-\delta)$. For such choice of d and w, when both v and m are sufficiently large, we have $\frac{t}{n} = \frac{1}{2}(1-\delta)+O\left(\max\left(\frac{1}{m}, \frac{1}{v}\right)\right)$, which approaches $1/2$, the optimal fraction of players a passive adversary can corrupt for a multiplicative secret sharing scheme. Alternatively, we may also set $w = \frac{m}{2}-1$ and $d = v$. Then if $v+1 \geq \frac{1}{\delta}$, we may also have $\frac{t}{n} \geq \frac{1}{2}-\delta$, providing an asymptotically optimal multiplicative instance of Σ.

Parameters for Strong-Multiplicativity. For Σ to be t-strongly multiplicative, we need $t \leq (m-2w)(v+1)+2d-1$. Note that if $w \geq \frac{m}{2}+1$, for a sufficiently large v, the upper bound $(m-2w)(v+1)+2d-1$ is negative. Hence, for a positive value of t, we need $w \leq \frac{m}{2}+1 < m$ for a sufficiently large m. Hence, as before, for Σ to be t-strongly multiplicative, we have $t = (d-1)(w+1)$. For any choice of $\delta > 0$, we may set $d \approx 1+\frac{(1-\delta)w(v+1)-1}{w+3} < 1+(1-\delta)(v+1)$ which is at most v assuming $v \geq \frac{2-\delta}{\delta}$. Furthermore, we may set $w \approx \frac{m}{3-\delta}$. For such choice of d and w, when v and m are sufficiently large, we have $\frac{t}{n} = \frac{1}{3}(1-\delta)+O\left(\frac{1}{mv}\right)$, which approaches $1/3$, the optimal number of corrupted players for a strongly-multiplicative secret sharing scheme.

3.2 Privacy Analysis

We have already determined under which choice of parameters our LSSS satisfies r-reconstruction, multiplicativity and strong multiplicativity. Another crucial aspect of an LSSS is its privacy threshold, that is, how many shares can an adversary know in such a way that they do not leak anything about the underlying secret. As we have mentioned, our scheme is a secret-sharing-based interpretation of the codes from [69], but in that work the authors did not consider privacy, and hence did not analyze this property. As it turns out, determining the privacy level of this scheme is not an easy task.

To better understand the complications of analyzing the privacy of our LSSS, we first perform a simple analysis that turns out to be far from what we can actually achieve. Consider an adversary that sees shares associated with a subset A of $\bigcup_{i=1}^{m} \beta_i H = \{\gamma_1, \ldots, \gamma_n\}$, and let us denote by A_i the intersection $A \cap \beta_i H$. Consider sharings $(s_1, \ldots, s_n)$ of a secret $s \in \mathbb{F}_q$, that is, $s_\ell = h(\gamma_\ell) = \sum_{i=0}^{d-1} \left(\sum_{j=0}^{w} b_{i,j} g(\gamma_\ell)^j \right) \gamma_\ell^i$. For every $\gamma_\ell \in A$ the adversary learns s_ℓ, but suppose temporarily that the adversary actually learns the "inner summands" $\{\sum_{j=0}^{w} b_{i,j} g(\gamma_\ell)^j\}_{i=0}^{d-1}$, which is *more information* than what is actually leaked to the adversary. Since $g(\mathtt{X})$ is constant in all of A_ℓ, we have that for every $\gamma \in A_\ell$: $\sum_{j=0}^{w} b_{i,j} g(\gamma)^j = \sum_{j=0}^{w} b_{i,j} g(\beta_\ell)^j$, which is a random polynomial of degree $\leq w$ in $g(\beta_\ell)$.

If we denote by A_{non} the set $\{i \in [m] : A_i \neq \emptyset\}$ (which corresponds to the amount of "groups" $\{\beta_i H\}_i$ for which the adversary has *at least* one share) we see that the only information the adversary sees is $|A_{non}|$ evaluations of each of the random degree-w polynomials $\sum_{j=0}^{w} b_{i,j} \mathtt{X}^j$, for $i = 0, \ldots, d-1$. If it happens to be the case that $|A_{non}| \leq w$, each of these evaluations leak nothing about $\sum_{j=0}^{w} b_{i,j} g(0)^j$, so in particular they leak nothing about the secret $s = \sum_{j=0}^{w} b_{0,j} g(0)^j$. One way in which it can happen that $|A_{non}| \leq w$ is if $|A| \leq w$ to begin with, which shows that, if the adversary corrupts at most w parties, then the shares of the corrupted parties leak nothing about the underlying secret. In other words, our LSSS has w-privacy, or equivalently, its privacy level is *at least* w.

The lower bound of w on the privacy level of our scheme obtained above is relatively easy to derive, but it is unfortunately too pessimistic. One way to see why this should be the case is by noticing that $|A| \leq w$ is a sufficient condition for $|A_{non}| \leq w$, but it is far from being necessary. It could be the case that $|A_{non}| \leq w$ in spite of $|A|$ being much larger than w, for example, if the adversary gets unlucky and all of his corrupted parties happen to be randomly assigned to the same coset (recall the adversary is static and the random assigments are done after the corrupted parties are set). Furthermore, in our analysis above we assumed that the adversary got $\{\sum_{j=0}^{w} b_{i,j} g(\gamma_\ell)^j\}_{i=0}^{d-1}$ for every $\gamma_\ell \in A$, while in reality he does not get these individual terms but rather the sum $s_\ell = \sum_{i=0}^{d-1} \sum_{j=0}^{w} b_{i,j} g(\gamma_\ell)^j \gamma_\ell^i$.

In what follows we perform a more extensive and accurate analysis that takes into account the observations above, together with several other extra considerations. As we will see, we are able to obtain the following theorem.

Theorem 2. *Let $\mathbb{F}_q$ be a finite field of q elements, v, m, w, d be positive integers such that $v+1$ divides $q-1, m \leq \frac{q-1}{v+1}, w \leq m-1$ and $d \leq v$. The secret sharing scheme Σ constructed in Construction 1 using the parameters q, v, n, m, d, w has t-privacy where*

$$t = \begin{cases} md-1, & \text{if } w = m-1 \\ (d-1)(w+1), & \text{otherwise} \end{cases}.$$

We see then that the privacy threshold can be actually lower bounded by $\approx w \cdot d$, which is around d times better than the pessimistic lower bound of w we obtained previously. In fact, as we will see later in Proposition 7, the privacy level of our construction can also be upper bounded by $\approx w \cdot d$, which shows that our improved analysis is closer to being optimal (however, there is still a *constant* gap between the lower and the upper bounds).

We will prove Theorem 2 in multiple steps. First, we provide a supporting proposition that is essential in the analysis of the privacy level for our LSSS.

Proposition 4 *Let $A \subseteq \bigcup_{i=1}^{m} \beta_i H$ of size at most $d(w+1)-1$ and $A_i = A \cap \beta_i H$. Without loss of generality, we assume $|A_1| \geq |A_2| \geq \cdots \geq |A_m|$. Then there exists a non-negative integer $M \leq w$, a vector $\boldsymbol{u} \in \mathbb{F}_q^{d(w-M+1)}$ and a matrix $\mathcal{A} \in \mathbb{F}_q^{(\sum_{i=M+1}^{m} |A_i|) \times (d(w-M+1))}$ satisfying the following: in order to show that the shares of A contain no information on the secret, it is sufficient to show that $\boldsymbol{u}$ does not belong to the row span of $\mathcal{A}$. More specifically, if $\boldsymbol{u}$ does not belong to the row span of $\mathcal{A}$, the distribution of the shares of A is independent of the secret.*

Proof. Each $\alpha \in A_i$ provides one evaluation point to the polynomial $f(\mathtt{X})|_{\beta_i H}$ which has degree $d-1$. Hence if $|A_i| \geq d$, the shares in A_i can be used to recover the values of $\sum_{j=0}^{w} a_{k,j} g(\beta_k)^j$ for $k = 0, \ldots, d-1$. Assuming that there are M different values of i such that $|A_i| \geq d$, this information provides M different evaluation points to the polynomial $f_k(\mathtt{Y}) \triangleq \sum_{j=0}^{w} a_{k,j} \mathtt{Y}^j$ for $k = 0, \ldots, d-1$. By assumption, we have $|A| \leq (w+1)d - 1$. Hence we must have $M \leq w$.

Since we want to show that shares of A contain no information on the secret, we want to find the values of $b_{i,j}$ for $i = 0, \ldots, d-1$ and $j = 0, \ldots, w$ such that the polynomial $h(\mathtt{X}) = \sum_{i=0}^{d-1} \left(\sum_{j=0}^{w} b_{i,j} g^j(\mathtt{X}) \right) \mathtt{X}^i$ satisfies $h(0) = 1$ and $h(\gamma) = 0$ for any $\gamma \in A$. For $k = 0, \ldots, d-1$, we denote by $h_k(\mathtt{X}) \triangleq \sum_{j=0}^{w} b_{k,j} \mathtt{X}^j$. Note that for the M distinct values of i such that $|A_i| \geq d$, we have $h_k(g(\beta_i)) = 0$ for $k = 0, \ldots, d-1$. Hence we have $(\mathtt{Y} - g(\beta_i)) \mid h_k(\mathtt{Y})$ for any of such i and $k = 0, \ldots, d-1$. So $\prod_{i=1}^{M} (\mathtt{Y} - g(\beta_i)) \mid h_k(\mathtt{Y})$. For $k = 0, \ldots, d-1$, we denote by $H_k(\mathtt{Y}) \triangleq \frac{h_k(\mathtt{Y})}{\prod_{i=1}^{M} (\mathtt{Y} - g(\beta_i))}$. Note that to find the values of $b_{k,j}$, it is equivalent to find $H_k(\mathtt{Y}) \triangleq \sum_{j=0}^{w-M} B_{k,j} \mathtt{Y}^j$ for $k = 0, \ldots, d-1$ such that $H_0(g(0)) = 1$ and $\sum_{k=0}^{d-1} H_k(g(\gamma)) \gamma^k = 0$ for any $\gamma \in \bigcup_{i=M+1}^{m} A_i$.

These new requirements can be represented as a problem of finding a solution of the matrix equation $\mathcal{M}\mathtt{X} = \boldsymbol{y}$. Here $\mathtt{X} \in \mathbb{F}_q^{d(w-M+1)}$ is defined as

$$\boldsymbol{x} = (B_{0,0}, B_{0,1}, \ldots, B_{0,w-M}, \ldots, B_{d-1,0}, \ldots, B_{d-1,w-M})^T$$

while $\boldsymbol{y}$ is a vector of length $1 + \sum_{i=M+1}^{m} |A_i|$ defined as $\boldsymbol{y} = (1, 0, 0, \ldots, 0)^T$. Lastly, $\mathcal{M}$ is a matrix with $1 + \sum_{i=M+1}^{m} |A_i|$ rows and $d(w-M+1)$ columns corresponding to the requirements defined by the system of equations where

$$\mathcal{M} = \begin{bmatrix} \boldsymbol{u} \\ \mathcal{A} \end{bmatrix} \in \mathbb{F}_q^{(1+\sum_{i=M+1}^{m} |A_i|) \times (d(w-M+1)))}.$$

Here $\boldsymbol{u}$, the first row of $\mathcal{M}$, corresponds to the equation related to the secret, i.e. $\sum_{j=0}^{w-M} b_{0,j} g(0)^j = 1$. The remaining rows of $\mathcal{M}$ corresponds to the shares of $\bigcup_{i=M+1}^{m} A_i$. More specifically, for any $\gamma \in A_i = A \cap \beta_i H \subseteq \bigcup_{i=M+1}^{m} A_i$, the row of $\mathcal{A}$ corresponding to γ is the following vector of length $d(w-M+1)$:

$$\mathcal{A}_\gamma = \left(1 \cdot \boldsymbol{g}(\beta_i) \| \gamma \cdot \boldsymbol{g}(\beta_i) \| \ldots \| \gamma^{d-1} \cdot \boldsymbol{g}(\beta_i)\right)$$

where $\boldsymbol{g}(\beta_i) = (1, g(\beta_i), g^2(\beta_i), \ldots, g^{w-M}(\beta_i))$.

We claim that if $\boldsymbol{u}$ does not belong to the row span of $\mathcal{A}$, then the required vector $\boldsymbol{x}$ exists, proving that the shares of A contain no information on the secret. Indeed, if $\boldsymbol{u}$ does not belong to the row span of $\mathcal{A}$, then there exists $\boldsymbol{z}$ that belongs to the dual of the row span of $\mathcal{A}$ such that $\boldsymbol{u} \cdot \boldsymbol{z} \neq 0$. Hence by using an appropriate scalar multiplication, we can find $\boldsymbol{x}$ belonging to the dual of the row span of $\mathcal{A}$ such that $\boldsymbol{u} \cdot \boldsymbol{x} = 1$ as required. □

Now we are ready to prove Theorem 2.

Proof (of Theorem 2). Let $A \subseteq \bigcup_{i=1}^{m} \beta_i H$ of size t and $A_i = A \cap \beta_i H$. Without loss of generality, we assume that $|A_1| \geq |A_2| \geq \cdots \geq |A_m|$. By Proposition 4, in order to show that the shares of A do not provide any information on the secret, it is sufficient to show that the vector $\boldsymbol{u} \triangleq (1, g(0), g^2(0), \ldots, g^{w-M}(0), 0, \ldots, 0) \in \mathbb{F}_q^{d(w-M+1)}$ does not belong to the span of the set

$$\left\{ \begin{array}{c} \mathcal{A}_\gamma \triangleq (1, g(\beta_i), \ldots, g^{w-M}(\beta_i), \gamma \cdot 1, \ldots, \gamma \cdot g^{w-M}(\beta_i), \\ \ldots, \gamma^{d-1} \cdot 1, \ldots, \gamma^{d-1} \cdot g^{w-M}(\beta_i)) : \\ i = M+1, \ldots, m, \gamma \in A_i \end{array} \right\}.$$

We prove this by contradiction. Suppose that there exist $\lambda_\gamma \in \mathbb{F}_q$ for $\gamma \in A$ such that $\boldsymbol{u} = \sum_{\gamma \in A} \lambda_\gamma \mathcal{A}_\gamma$. Since the first entry of $\boldsymbol{u}$ is 1, we must have λ_γ to not all be zero. This shows that the set

$$\left\{ \begin{array}{c} \left(\gamma \cdot 1, \ldots, \gamma \cdot g^{w-M}(\beta_i), \ldots, \gamma^{d-1} \cdot 1, \ldots, \gamma^{d-1} \cdot g^{w-M}(\beta_i)\right) \\ : i = M+1, \ldots, m, \gamma \in A_i \end{array} \right\}$$

is not linearly independent. More specifically, for $j = 1, \ldots, d-1$, we have

$$(0, \ldots, 0) = \sum_{i=M+1}^{m} \left(\sum_{\gamma \in A_i} \lambda_\gamma \gamma^j \right) \cdot \left(1, \ldots, g^{w-M}(\beta_i)\right).$$

For each j, this defines a homogeneous system of linear equation that can be represented as a matrix equation with the matrix being a Vandermonde-like matrix with $w - M + 1$ rows and $m - M$ columns. Now we divide the argument into two cases based on the value of w.

1. When $w = m - 1$, each matrix is a square and hence invertible. This implies that for any $j = 1, \ldots, d-1$ and $i = M+1, \ldots, m$, we have $\sum_{\gamma \in A_i} \lambda_\gamma \gamma^j = 0$. For each $i = M+1, \ldots, m$, we have $\sum_{\gamma \in A_i} \lambda_\gamma \gamma^j = 0$ for $j = 1, \ldots, d-1$. This again defines a homogeneous system of linear equations with $|A_i|$ unknowns and $d-1$ equations with the corresponding matrix being a Vandermonde-like matrix. By the argument provided in the proof of Proposition 4, we have $|A_i| \leq d - 1$. Hence for each $i = M + 1, \ldots, m$, the system is overdefined. Because of this, we must have $\lambda_\gamma = 0$ for any $\gamma \in A_i$ for any $i = M+1, \ldots, m$. However, this is a contradiction with the earlier observation that λ_γ cannot be all zero. This shows that if $w = m - 1$, for any $A \subseteq \bigcup_{i=1}^{m} \beta_i H$ such that $|A| = md-1$, the shares of A do not contain any information about the secret, proving that Σ provides at least $md - 1$ privacy when we set $w = m - 1$.
2. Next, we assume that $w < m - 1$. Recall that we assumed the existence of $(\lambda_\gamma)_{\gamma \in A}$ that are not identically zero such that for any $j = 1, \ldots, d-1$ and $\ell = 0, \ldots, w-M$, we have $\sum_{i=M+1}^{m} \left(\sum_{\gamma \in A_i} \lambda_\gamma \gamma^j \right) g^\ell(\beta_i) = 0$. Note that this defines a homogeneous system of linear equations with $(d-1)(w-M+1)$ equations and $\sum_{i=M+1}^{m} |A_i| = t - \sum_{i=1}^{M} |A_i| \leq t - Md$ unknowns. By the choice of t, we have $(d-1)(w-M+1) \geq t - Md$.

   We derive a contradiction by showing that this system of linear equations can only have a zero solution, which is a contradiction to the assumption that $(\lambda_\gamma)_{\gamma \in A}$ are not all zero. We define the matrix $\mathcal{N}$ over $\mathbb{F}_q$ with $(d-1)(w-M+1)$ rows and $\sum_{i=M+1}^{m} |A_i|$ columns, which corresponds to the system of linear equations discussed above. For the matrix $\mathcal{N}$, we group the rows to $w-M+1$ distinct groups based on the value of ℓ. This means that each group consists of $d-1$ rows with various values of $j \in \{1, \ldots, d-1\}$.

   We further note that each column of $\mathcal{N}$ corresponds to some $\gamma \in A$. Based on this correspondence, we further group the columns of $\mathcal{N}$ to $m - M$ column groups based on the cosets $\beta_i H$ that contains the corresponding γ. By using this grouping method, we can represent $\mathcal{N}$ as a block matrix. For $1 \leq i \leq w - M + 1$ and $1 \leq j \leq m - M$ we denote by $\mathcal{N}_{i,j}^{(1)}$ the block matrix belonging to the i-th row group and j-th column group. More specifically, for $\ell = 1, \ldots, d-1$, the ℓ-th row of $\mathcal{N}_{i,j}^{(1)}$ is $\left[\left(\gamma^\ell g^{i-1}(\beta_{j+M}) \right)_{\gamma \in \beta_{j+M} H} \right]$. Hence

$\mathcal{N}^{(1)}_{i,j}$ can be written as

$$\mathcal{N}^{(1)}_{i,j} = g^{i-1}(\beta_{j+M}) \cdot \begin{bmatrix} (\gamma)_{\gamma \in \beta_{j+M} H} \\ (\gamma^2)_{\gamma \in \beta_{j+M} H} \\ \vdots \\ \left(\gamma^{d-1}\right)_{\gamma \in \beta_{j+M} H} \end{bmatrix}.$$

So $\mathcal{N}^{(1)}_{i,j}$ is a non-zero multiple of a Vandermonde-like matrix with $d-1$ rows and $|A_{j+M}| \leq d-1$ columns. Hence any square sub-matrix of $\mathcal{N}^{(1)}_{i,j}$ is invertible. Our argument is based on the following claim that can be easily verified.

Claim 5 *Suppose that $\mathcal{N}$ has an alternative representation as a block matrix $(\mathcal{N}_{i,j})$ with more row groups than column groups such that $\mathcal{N}_{i,i}$ is an invertible square matrix for all i. Then the only solution of $\mathcal{N}\boldsymbol{x} = \mathbf{0}$ is $\boldsymbol{x} = \mathbf{0}$.*

By Claim 5, to derive the contradiction, it is sufficient to construct such alternative representation $(\mathcal{N}_{i,j})$ of $\mathcal{N}$ as a block matrix satisfying the requirement provided. We will keep the partition of the column as before. However, we will partition the row following the number of columns in each column group. More specifically, we define the first row group to be the first $|A_{m+1}|$ rows of $\mathcal{N}$, the second row group to be the next $|A_{m+2}|$ rows of $\mathcal{N}$ and so on while the $(m-M+1)$-st row group to contain the remaining rows. Note that this can always be done since $\mathcal{N}$ has more rows than columns. Furthermore, since $|A_i| \leq d-1$ for any $i = M+1, \ldots, m$, there exists a value $\ell = 1, \ldots, w-M+1$ such that the square submatrix $\mathcal{N}_{i,i}$ has all its rows to belong to $\mathcal{N}^{(1)}_{i,\ell}$ and possibly $\mathcal{N}^{(1)}_{i,\ell+1}$. So there exists $s, u \in \{1, \ldots, d-1\}$ such that

$$\mathcal{N}_{i,i} = \left[\begin{array}{c} (g^{i-1}(\beta_{i+M})\gamma^s)_{\gamma \in \beta_{i+M} H} \\ (g^{i-1}(\beta_{i+M})\gamma^{s+1})_{\gamma \in \beta_{i+M} H} \\ \vdots \\ (g^{i-1}(\beta_{i+M})\gamma^{d-1})_{\gamma \in \beta_{i+M} H} \\ \hline (g^{i}(\beta_{i+M})\gamma)_{\gamma \in \beta_{i+M} H} \\ (g^{i}(\beta_{i+M})\gamma^2)_{\gamma \in \beta_{i+M} H} \\ \vdots \\ (g^{i}(\beta_{i+M})\gamma^{u})_{\gamma \in \beta_{i+M} H} \end{array}\right].$$

Recall that $\mathcal{N}_{i,i}$ is a square matrix. Hence, we can partition $\beta_{i+M}H$ to two sets L_i, R_i such that $|L_i| = d - s$ and $|R_i| = u$. Hence we can rewrite $\mathcal{N}_{i,i}$ to

$$\mathcal{N}_{i,i} = \left[\begin{array}{c|c} (g^{i-1}(\beta_{i+M})\gamma^s)_{\gamma\in L_i} & (g^{i-1}(\beta_{i+M})\gamma^s)_{\gamma\in R_i} \\ (g^{i-1}(\beta_{i+M})\gamma^{s+1})_{\gamma\in L_i} & (g^{i-1}(\beta_{i+M})\gamma^{s+1})_{\gamma\in R_i} \\ \vdots & \vdots \\ (g^{i-1}(\beta_{i+M})\gamma^{d-1})_{\gamma\in L_i} & (g^{i-1}(\beta_{i+M})\gamma^{d-1})_{\gamma\in R_i} \\ \hline (g^{i}(\beta_{i+M})\gamma)_{\gamma\in L_i} & (g^{i}(\beta_{i+M})\gamma)_{\gamma\in R_i} \\ (g^{i}(\beta_{i+M})\gamma^2)_{\gamma\in L_i} & (g^{i}(\beta_{i+M})\gamma^2)_{\gamma\in R_i} \\ \vdots & \vdots \\ (g^{i}(\beta_{i+M})\gamma^{u})_{\gamma\in L_i} & (g^{i}(\beta_{i+M})\gamma^{u})_{\gamma\in R_i} \end{array}\right].$$

It is easy to see that the top left and bottom right sub-matrices of $\mathcal{N}_{i,i}$ are square matrices that can be rewritten as non-zero constant multiples of a Vandermonde-like matrix. Hence by Claim 5, $\mathcal{N}_{i,i}$ is also invertible for all $i = M+1, \ldots, m$. We can use Claim 5 again to conclude that we must have $(\lambda_\gamma)_{\gamma\in A}$ to be identically zero, contradicting the assumption that it cannot be a zero vector, completing the proof for the privacy level of the secret sharing scheme defined in Construction 1 when $w < m - 1$. □

Remark 6 *In general, the privacy threshold that is claimed in Theorem 2 is not guaranteed to be the largest possible privacy threshold of the scheme. On the other hand, Proposition 7 below provides an upper bound on the largest possible privacy threshold of the scheme. So in particular, when $w = m - 1$, the privacy threshold provided in Theorem 2 is the largest possible privacy threshold of the secret sharing scheme Σ proposed in Construction 1, and when $w < m - 1$, the lower and upper bounds are off by a constant term.*

Proposition 7 *If the secret sharing scheme constructed in Construction 1 provides t-privacy, then $t \leq d(w+1) - 1$.*

Proof. Set $A \subseteq \bigcup_{i=1}^{w+1} \beta_i H$ such that for each $i = 1, \ldots, w+1$, we have $|A_i| = d$. Since each $\gamma \in A_i$ provides an evaluation point to the polynomial $f(\mathtt{X})|_{\beta_i H}$ which has degree $d-1$, the shares of A_i can be used to recover the value of $\sum_{j=0}^{w} a_{k,j} g^j(\beta_k)$ for $k = 0, \ldots, d-1$. In particular, it provides an evaluation point of $\sum_{j=0}^{w} a_{0,j} y^j$ where the evaluation point is $y = g(\beta_i)$. Note that this is a polynomial of degree w and we are given the value of the polynomial in $w+1$ evaluation points which are pairwise distinct by Lemma 3. Hence, this information can be used to recover the polynomial $\sum_{j=0}^{w} a_{0,j} y^j$, which can then be used to calculate $\sum_{j=0}^{w} a_{0,j} g^j(0) = S$, proving that if we allow $A \geq d(w+1)$, it is possible for such set to not only learn some information about the secret but also fully recover it. Hence, if the secret sharing scheme provides t-privacy, we must have $t \leq d(w+1) - 1$. □

4 Passively Secure Repairing Protocol for Multiplicative Variants of Σ

In this section, we discuss the secure repairing capability of multiplicative instances of Σ in the scenario that a set of τ players is corrupted by a semi-honest adversary. Here we set $w-1 \leq \frac{m}{2}$ so that multiplicativity is achieved, and we require τ, the number of corrupted parties, to be upper-bounded by the privacy level of the secret sharing scheme, which, from Theorem 2, equals $(d-1)(w+1)$ (since $w < m-1$).

In this section we are interested in different metrics/properties such as privacy of the secret after some shares have been repaired, the number of players contacted during the repairing process and the bandwidth required for the repairing process. First we restate the repairable claim as stated in Theorem 3. We note that for simplicity of discussion, we consider one of the instantiations where we set $d = v$ and $w = \frac{m}{2} - 1$.

Theorem 3. *Consider the secret sharing scheme Σ presented in Construction 1 with $d = v, w \leq \frac{m}{2} - 1, r = w(v+1) + v$ and $t = (v-1)(w+1)$. Without loss of generality, suppose that the player P_1 identified by γ_1 loses his share $f(\gamma_1)$. Then he can recover the value of $f(\gamma_1)$ by contacting v other players. The repairing process requires the v contacted players to send in total $2v \log q$ bits of data to each other and P_1, while P_1 needs to send $v \log q$ bits of data to the v contacted players. Furthermore, assuming that $v = \ln^{(1+\varepsilon)} n$ for some $\varepsilon > 0$, and that the adversary corrupts only t parties, such repair scheme does not leak any information about either the secret or the shares of the contacted players except with negligible probability in n.*

Proof. Assume that $\gamma_1 = \beta_1 \alpha_1$ for some $\alpha_1 \in H$. Consider $f(\mathtt{X})|_{\beta_1 H}$, the restriction of $f(\mathtt{X})$ to $\beta_1 H$. It is easy to see that $f(\mathtt{X})|_{\beta_1 H} = \sum_{i=0}^{v-1} \left(\sum_{j=0}^{w} a_{i,j} g(\beta_1)^j \right) \mathtt{X}^i$, which is a polynomial of degree at most $v-1$. Label the other v elements in $\beta_1 H$ by $\gamma_2, \ldots, \gamma_{v+1}$ and the $v+1$ corresponding players by $P_1, \ldots, P_{v+1}$. First, these $v+1$ players execute a simple multiparty computation protocol to generate a random shared mask, denoted by $\mathsf{MaskGen}$. Its full specification can be found in Algorithm 2.

Algorithm 2 Random Mask Generation $(h(\gamma_1), \ldots, h(\gamma_{v+1})) \leftarrow \mathsf{MaskGen}()$

1: **for** $i = 1, \ldots, v+1$ **do**
2: Player P_i randomly selects a polynomial $h_i(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]_{<v}$ of degree at most $v-1$ and sends $h_i(\gamma_j)$ to player P_j for $j = 1, \ldots, v+1$;
3: Upon receiving $h_j(\gamma_i)$ for $j = 1, \ldots, v+1$, player P_i defines $h(\gamma_i) = \sum_{j=1}^{v+1} h_j(\gamma_i)$;
4: **end for**
5: The generated mask is $h(x)$, which is shared as $(h(\gamma_1), \ldots, h(\gamma_{v+1}))$;

Note that since h has degree $\leq v-1$, if the adversary only corrupts up to $v-1$ of the $v+1$ players, he learns no information about the polynomial $h(\mathtt{X})$. Now

we define the repairing process conducted by $P_1, \ldots, P_{v+1}$ to repair the share of P_1. We denote such algorithm by Repair, which can be found in Algorithm 3.

Algorithm 3 Share Repair $(f(\gamma_1), -, \ldots, -) \leftarrow \mathsf{Repair}(-, f(\gamma_2), \ldots, f(\gamma_{v+1}))$

1: The $v+1$ players execute MaskGen to generate a random mask $(h(\gamma_1), \ldots, h(\gamma_{v+1}))$ where $h(\gamma_i)$ is held by P_i;
2: **for** $i = 2, \ldots, v+1$ **do**
3: P_i calculates $f(\gamma_i)|_{\beta_1 H} + h(\gamma_i)$ and sends it to P_1;
4: **end for**
5: Since $f(\mathtt{X})|_{\beta_1 H} + h(\mathtt{X})$ is a polynomial of degree at most $v-1$ and P_1 obtains v of its evaluation points, P_1 can recover the polynomial and in particular, she can recover $f(\gamma_1)|_{\beta_1 H} + h(\gamma_1)$;
6: Since P_1 knows $h(\gamma_1)$, she can then recover $f(\gamma_1)$;

It is easy to see that Algorithm 3 correctly recovers $f(\gamma_1)$ for P_1 when the adversary is semi-honest. Now we analyze in more detail that the privacy of the secret and the privacy of the shares of honest parties is maintained. Let A_1 be the set of corrupted parties in $\beta_1 H$. Note that since $P_2, \ldots, P_{v+1}$ receive no additional information from the execution of Repair, a possible leak of information is only possible if $P_1 \in A_1$. Note that since Repair only involves players in the same coset, if $A_1 = \beta_1 H$, then the amount of information that the adversary learns will not change after the execution of Repair. Given this, we can assume that $|A_1| \leq v$.

First, suppose that $|A_1| \leq v-1$. In this case, P_1 learns the polynomial $f|_{\beta_1 H} + h$ along with at most $v-1$ evaluation points of $h(x)$. Since the degree of $h(\mathtt{X})$ is $v-1$, the adversary learns no information on $h(\gamma_i)$ for $\gamma_i \notin A_1$. So aside from the information about $f(\gamma_1)$, the adversary learns no further information about the shares of the other honest parties. This also implies that the repairing process does not increase the amount of information the adversary has on the other shares. So as long as the total number of corrupted parties in A_1 is at most $v-1$, no information on the secret is leaked from the execution of Repair.

From the above, we see that leakage on either the shares of the honest parties or the secret, when repairing P_1's share, is possible only if $|A_1| = v$. More generally, other parties might be repaired during a protocol execution, so leakage is only possible if there exists i such that $|A_i| = v$. The rest of this proof is devoted to showing that this event happens with low probability, taking as a starting point the fact that the adversary is assumed static and the random assignments of the n field elements is performed after the corruptions have been established.

Let $\mathcal{E}$ be the event that there exists at least one i such that $|A_i| = v$, and let $\mathcal{E}^c$ be the event that $|A_i| \neq v$ for every $i = 1, \ldots, m$. We aim to show that $\Pr(\mathcal{E})$ is negligible. We define $\mathcal{E}_i$ to be the event that $|A_i| = v$. It is easy to see that $P(\mathcal{E}_i) = \frac{\binom{t}{v} \cdot (n-t)}{\binom{m(v+1)}{v+1}}$. We assume that as n goes to infinity, we also have

both m and v to also approach infinity. Then, by union bound, $\Pr(\mathcal{E})$ is at most $m(n-t)\frac{\binom{t}{v}}{\binom{m(v+1)}{v+1}} \leq n^2\frac{\binom{t}{v}}{\binom{n}{v}}$.

Recall that for any integers $0 < b < a$, we have the following approximation by the use of Stirling's approximation,

$$\binom{a}{b} \approx \sqrt{\frac{a}{2\pi b(a-b)}}\frac{a^a}{b^b(a-b)^{a-b}} = \frac{1}{\sqrt{2\pi b}}\sqrt{\frac{a}{a-b}}\left(\frac{a}{a-b}\right)^a\left(\frac{a-b}{b}\right)^b.$$

Then if $b = o(a)$, there exist two positive constants $0 < \lambda_1 < \lambda_2$ such that $\lambda_1 < \sqrt{\frac{a}{2\pi(a-b)}} < \lambda_2$. Hence we have that if $b = o(a)$, $\frac{\lambda_1}{\sqrt{b}}\left(\frac{a}{a-b}\right)^a\left(\frac{a-b}{b}\right)^b < \binom{a}{b} < \frac{\lambda_2}{\sqrt{b}}\left(\frac{a}{a-b}\right)^a\left(\frac{a-b}{b}\right)^b$.

Assuming that w also approaches infinity as n does, $\Pr(\mathcal{E})$ can be upper bounded by $\frac{\lambda_2}{\lambda_1}n^2\left(\frac{t-v}{n-v}\right)^v\left(\frac{t}{t-v}\right)^t\left(\frac{n-v}{n}\right)^n$

Hence $\ln(\Pr(\mathcal{E})) \leq \ln\left(\frac{\lambda_2}{\lambda_1}\right) + 2\ln n + v\ln\frac{t-v}{n-v} + t\ln\left(1+\frac{v}{t-v}\right) + n\ln\left(1-\frac{v}{n}\right)$. Note that for any $v > 0$, we have $\frac{t-v}{n-v} < \frac{t}{n}$. We denote $\theta = \frac{t}{n}$, which is a constant that is smaller than 1. Recall that for $|x| < 1$, we have $\ln(1+x) = \sum_{i\geq 1}(-1)^{i+1}\frac{x^i}{i}$. Since $v = o(t)$ and $t < n$, we can approximate $\ln\left(1+\frac{v}{t-v}\right) = \frac{v}{t-v} + O\left(\left(\frac{v}{t-v}\right)^2\right) \approx \frac{v}{t}$ and $\ln\left(1-\frac{v}{n}\right) = -\frac{v}{n} + O\left(\left(\frac{v}{n}\right)^2\right) \approx -\frac{v}{n}$.

Hence we have $\ln(\Pr(\mathcal{E})) \leq \ln\left(\frac{\lambda_2}{\lambda_1}\right) + 2\ln n + v\ln\theta$. By assumption, since $v = \Omega(\ln^{(1+\varepsilon)} n)$, we have $\ln(\Pr(\mathcal{E})) = -\Omega\left(\ln^{(1+\varepsilon)} n\right)$, which implies that $\Pr(\mathcal{E}) = e^{-\Omega\left(\ln^{(1+\varepsilon)} n\right)}$ which is negligibly small when n grows sufficiently large. □

5 Actively Secure Repairing Protocol for Strongly-Multiplicative Variants of Σ

In this section, we discuss the repairability of a strongly-multiplicative instance of Σ in the scenario that a set of τ players is corrupted by an active adversary. As discussed before, to achieve strong multiplicativity, we must have $w < m - 1$. Furthermore, similar to the discussion in Section 4, we require τ to be at most the privacy threshold, which equals $(d-1)(w+1)$. In general, our actively secure repairing process is very similar to the passively secure process defined in Section 4. The crucial difference here is the need of the participants to perform verification process to the generated random mask which is done through the protocol described in [5]. We restate and prove the repairable claim for Σ, as stated in Theorem 4.

Theorem 4. *Let Σ be the secret sharing scheme constructed in Algorithm 1 with $w \approx \frac{m}{3}, r = w(v+1)+d$ and $t \approx \frac{1}{3}n$. Without loss of generality, suppose that the*

player P_1 identified by γ_1 loses his share $f(\gamma_1)$. Then he can recover the value of $f(\gamma_1)$ by contacting v other players where each involved player needs to send $O(v \log q)$ bits of data to each other. Furthermore, assuming that $v = \Omega(\ln^{(1+\varepsilon)} n)$ for some $\varepsilon > 0$, such repair scheme is unconditionally secure with overwhelming probability with respect to both the secret and the shares of the players against an active adversary controlling up to t players.

Proof. We first start by defining two algorithms, which are analogues of MaskGen and Repair described in Theorem 3. We will use the notations described in the first paragraph of the proof of Theorem 3 except that $f(\mathtt{X})|_{\beta_1 H}$ is a polynomial of degree at most $d-1$ instead of $v-1$. The full specification of the protocol ActMaskGen, which is a variant of MaskGen, can be found in Algorithm 4. Here, we take $d = \frac{v+1}{3}$.

Algorithm 4 Actively Secure Random Mask Generation $(h(\gamma_1), \ldots, h(\gamma_{v+1})) \leftarrow$ ActMaskGen()

1: **for** $i = 1, \ldots, v+1$ **do**
2: Player P_i randomly selects a polynomial $h_i(\mathtt{X}) \in \mathbb{F}_q[\mathtt{X}]_{<\frac{v+1}{3}}$ of degree at most $\frac{v+1}{3} - 1$ and sends $h_i(\gamma_j)$ to player P_j for $j = 1, \ldots, v+1$;
3: Player P_i follows the verification process described in [5] to prove to the other players that $h_i(\gamma_j)$ is indeed an evaluation of some polynomial of degree at most $\frac{v+1}{3} - 1$;
4: If the verification protocol fails, the protocol aborts;
5: If the verification protocol is successful, having $h_j(\gamma_i)$ for $j = 1, \ldots, v+1$, player P_i defines $h(\gamma_i) = \sum_{j=1}^{v+1} h_j(\gamma_i)$;
6: **end for**
7: The generated mask is $h(x)$, which is shared as $(h(\gamma_1), \ldots, h(\gamma_{v+1}))$;

Note that since h has degree $\frac{v+1}{3} - 1$, if the adversary only corrupts up to $\frac{v+1}{3} - 1$ of the $v+1$ players, he learns no information about the polynomial $h(\mathtt{X})$. Furthermore, since such sharing can be seen as a codeword of a Reed-Solomon code of length $v+1$ and dimension $\frac{v+1}{3}$, it can correct up to $\frac{v+1}{3}$ errors. So in particular, it can correct any malicious behavior of an adversary controlling up to $\frac{v+1}{3} - 1$ of the $v+1$ players. Now we define the repairing process conducted by $P_1, \ldots, P_{v+1}$ to repair the share of P_1. We denote such algorithm by Repair, which can be found in Algorithm 3.

It is easy to see that Algorithm 3 correctly recovers $f(\gamma_1)$ for P_1 when the adversary is malicious and controls up to $\frac{v+1}{3} - 1$ out of the $v+1$ involved players. Now, we consider the security of the secret and the shares of honest participating parties against the active adversary. Let A_1 be the set of corrupted parties in $\beta_1 H$. Note that since $P_2, \ldots, P_{v+1}$ learns no additional information either from the execution of Repair, a possible leak of information is only possible if $P_1 \in A_1$. Note that since Repair only involves players in the same coset, if $A_1 = \beta_1 H$, the amount of information that the adversary learns will not change

Algorithm 5 Share Repair $(f(\gamma_1), -, -, \ldots, -) \leftarrow \mathsf{ActRepair}(-, f(\gamma_2), \ldots, f(\gamma_{v+1}))$

1: The $v+1$ players executes $\mathsf{MaskGen}$ to generate a random mask $(h(\gamma_1), \ldots, h(\gamma_{v+1}))$ where $h(\gamma_i)$ is held by P_i;
2: **for** $i = 2, \ldots, v+1$ **do**
3: P_i calculates $f(\gamma_i)|_{\beta_1 H} + h(\gamma_i)$ and sends it to P_1;
4: **end for**
5: Since $f(\mathtt{X})|_{\beta_1 H} + h(\mathtt{X})$ is a polynomial of degree at most $\frac{v+1}{3} - 1$ and P_1 obtains v of its evaluation points, P_1 can recover and correct the polynomial and in particular, she can recover $f(\gamma_1)|_{\beta_1 H} + h(\gamma_1)$ as long as the adversary only controls up to $\frac{v+1}{3} - 1$ out of the $v+1$ players.
6: Since P_1 knows $h(\gamma_1)$, she can then recover $f(\gamma_1)$;

after the execution of Repair. So we can assume that $|A_1| \leq v$. Suppose that $|A_1| \leq \frac{v+1}{3} - 1$. In this case, P_1 learns the polynomial $f|_{\beta_1 H} + h$ along with at most $\frac{v+1}{3} - 1$ evaluation points of $h(x)$. Since the degree of $h(x)$ is $\frac{v+1}{3} - 1$, the adversary learns no information on $h(\gamma_i)$ for $\gamma_i \notin A_1$. So aside from the information about $f(\gamma_1)$, the adversary learns no further information about the shares of the other honest parties. This also implies that the repairing process does not increase the amount of information the adversary has on the other shares. So as long as the total number of corrupted parties in each coset is at most $\frac{v+1}{3}$, no information on the secret is leaked from the execution of Repair. So leakage of information of either the share of honest parties or the secret is possible if there exists i such that $|A_i| \geq \frac{v+1}{3}$. So the probability that the execution of Repair leaks information about either the secret or the share of honest parties is at most the probability that during the random assignments of the n field elements, there exists $i \in [m]$ such that $\frac{v+1}{3} \leq |A_i| \leq v$.

Let $\mathcal{F}$ be the event that there exists at least one i such that $\frac{v+1}{3} \leq |A_i| \leq v$ and $\mathcal{F}^c$ be the event that $|A_i| < \frac{v+1}{3}$ for any $i = 1, \ldots, m$. We aim to show that $\Pr(\mathcal{F})$ is negligible. We define $\mathcal{F}_i$ to be the event that $\frac{v+1}{3} \leq |A_i| \leq v$. Then we have

$$\Pr(\mathcal{F}_i) = \frac{\sum_{\kappa=\frac{v+1}{3}}^{v} \binom{t}{\kappa}\binom{n-t}{v+1-\kappa}}{\binom{n}{v+1}}$$

Claim 8 *Let $0 < v < t < n$ such that $t \leq \frac{1}{3}n$ and $\kappa \in \left\{\frac{v+1}{3}, \frac{v+1}{3}+1, \ldots, v\right\}$. Then $\binom{t}{\kappa}\binom{n-t}{v+1-\kappa}$ achieves its maximum of $\binom{t}{\frac{v+1}{3}}\binom{n-t}{\frac{2(v+1)}{3}}$ when $\kappa = \frac{v+1}{3}$.*

Proof. Let $\kappa \geq \frac{v+1}{3}$. Then

$$\begin{aligned}\frac{\binom{t}{\kappa+1}\binom{n-t}{v-\kappa}}{\binom{t}{\kappa}\binom{n-t}{v+1-\kappa}} &= \frac{(t-\kappa)(v+1-\kappa)}{(\kappa+1)(n-t-v+\kappa)} \\ &\le \frac{\left(t-\frac{v+1}{3}\right)\left(\frac{2}{3}(v+1)\right)}{\left(\frac{v+1}{3}\right)\left(n-t-(v+1)+\frac{v+1}{3}\right)} \\ &\le 2\frac{(v+1)\left(\frac{1}{3}m-\frac{1}{3}\right)}{(v+1)\left(\frac{2}{3}m-\frac{2}{3}\right)} = 1.\end{aligned}$$

This shows that if $t \le \frac{1}{3}n$ and $\kappa \ge \frac{v+1}{3}$, the term $\binom{t}{\kappa}\binom{n-t}{v+1-\kappa}$ is decreasing. Hence, it achieves its maximum of $\binom{t}{\frac{v+1}{3}}\binom{n-t}{\frac{2(v+1)}{3}}$ when $\kappa = \frac{v+1}{3}$. □

Following the discussion in Section 3.1 regarding the parameter choices, for Σ to be t'-strongly multiplicative where t' is sufficiently close to $\frac{1}{3}n$, for $\delta \in (0,1)$ and sufficiently large m and v, we may choose $d \approx 1 + (1-\delta)(v+1) \le v$ and $w \approx \frac{m}{3-\delta}$. So we have $t \approx \frac{1}{3}(1-\delta)n$.

Then by union bound, we have the following upper bound

$$\Pr(\mathcal{F}) \le \frac{2n}{3}\frac{\binom{(v+1)(w+1)}{\frac{v+1}{3}}\binom{(m-w+1)(v+1)}{\frac{2(v+1)}{3}}}{\binom{m(v+1)}{v+1}}.$$

Recall that by the discussion in the proof of Theorem 3, if $b = o(a)$, there exist positive constants $0 < \lambda_1 < \lambda_2$ such that

$$\frac{\lambda_1}{\sqrt{b}}\left(\frac{a}{a-b}\right)^a\left(\frac{a-b}{b}\right)^b < \binom{a}{b} < \frac{\lambda_2}{\sqrt{b}}\left(\frac{a}{a-b}\right)^a\left(\frac{a-b}{b}\right)^b.$$

So we have the following upper bound

$$\begin{aligned}\Pr(\mathcal{F}) \le{}& \frac{2}{3}n\frac{\lambda_2^2}{\lambda_1}\sqrt{\frac{v+1}{\frac{v+1}{3}\frac{2(v+1)}{3}}}\frac{(w+1)^{(v+1)(w+1)}3^{\frac{v+1}{3}}}{\left(w+\frac{2}{3}\right)^{(v+1)\left(w+\frac{2}{3}\right)}} \\ &\cdot\left(\frac{3}{2}\right)^{\frac{2}{3}(v+1)}\cdot\frac{(m-w+1)^{(m-w+1)(v+1)}}{\left(m-w+\frac{1}{3}\right)^{(v+1)\left(m-w+\frac{1}{3}\right)}} \\ &\cdot\frac{(m-1)^{(m-1)(v+1)}}{m^{m(v+1)}} \\ ={}& \sqrt{2}\frac{n}{\sqrt{v+1}}\frac{\lambda_2^2}{\lambda_1}\cdot\frac{3^{v+1}}{2^{\frac{2}{3}(v+1)}}\cdot\left(\frac{w+1}{w+\frac{2}{3}}\right)^{\left(w+\frac{2}{3}\right)(v+1)} \\ &\cdot\left(\frac{m-1}{m}\right)^{(m-1)(v+1)}\cdot\frac{(w+1)^{\frac{v+1}{3}}(m-w+1)^{\frac{2}{3}(v+1)}}{m^{v+1}} \\ &\cdot\left(\frac{m-w+1}{m-w+\frac{1}{3}}\right)^{\left(m-w+\frac{1}{3}\right)(v+1)}\end{aligned}$$

Then for a sufficiently large m, we have

$$\left(\frac{w+1}{w+\frac{2}{3}}\right)^{\left(w+\frac{2}{3}\right)(v+1)} \cdot \left(\frac{m-1}{m}\right)^{(m-1)(v+1)} \cdot \left(\frac{m-w+1}{m-w+\frac{1}{3}}\right)^{\left(m-w+\frac{1}{3}\right)(v+1)} \leq 1+\delta.$$

Furthermore, for a sufficiently large m and v, we have

$$\frac{(w+1)^{\frac{v+1}{3}}(m-w+1)^{\frac{2}{3}(v+1)}}{m^{v+1}} \leq \frac{(2-\delta)^{\frac{2}{3}(v+1)}}{(3-\delta)^{v+1}}.$$

So the upper bound above can be simplified to the following

$$\Pr(\mathcal{F}) \leq \sqrt{2}(1+\delta)\frac{n}{\sqrt{v+1}}\frac{\lambda_2^2}{\lambda_1}\left(\frac{3}{3-\delta}\cdot\left(\frac{2-\delta}{2}\right)^{\frac{2}{3}}\right)^{v+1}.$$

Note that by the choice of δ, $\frac{3}{3-\delta}\cdot\left(\frac{2-\delta}{2}\right)^{\frac{2}{3}} < 1$. Hence, there exists $C > 0$ such that $\ln\left(\frac{3}{3-\delta}\cdot\left(\frac{2-\delta}{2}\right)^{\frac{2}{3}}\right) < -C$. Then we have

$$\ln\Pr(\mathcal{F}) \leq \frac{1}{2}\ln 2 + \ln(1+\delta) + \ln n - \frac{1}{2}\ln(v+1) + \ln\left(\frac{\lambda_2^2}{\lambda_1}\right) - C(v+1)$$

By assumption, $v+1 = \ln^{(1+\varepsilon)} n$. Then $\ln Pr(\mathcal{F}) = -\Omega(\ln^{(1+\varepsilon)} n)$. This implies that $\Pr(\mathcal{F}) = e^{-\Omega(\ln^{(1+\varepsilon)} n)}$ which is negligibly small when n is sufficiently large.

Acknowledgments

This research / project is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

The work of Chaoping Xing was supported in part by the National Key Research and Development Project under Grant 2022YFA1004900, in part by the National Natural Science Foundation of China under Grants 12031011, 12361141818 and 12271084.

This paper was prepared in part for information purposes by the Artificial Intelligence Research Group and the AlgoCRYPT CoE of JPMorgan Chase & Co and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy, or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer, or solicitation for

the purchase or sale of any security, financial instrument, financial product, or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

1. A. Agarwal and A. Mazumdar. Security in locally repairable storage. *IEEE Transactions on Information Theory*, 62(11):6204–6217, 2016.
2. S. Badrinarayanan, A. Jain, N. Manohar, and A. Sahai. Secure mpc: laziness leads to god. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 120–150. Springer, 2020.
3. S. B. Balaji and P. V. Kumar. A tight lower bound on the sub- packetization level of optimal-access msr and mds codes. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2381–2385, 2018.
4. C. Baum, D. Cozzo, and N. P. Smart. Using topgear in overdrive: a more efficient zkpok for spdz. In *International Conference on Selected Areas in Cryptography*, pages 274–302. Springer, 2019.
5. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 1–10, New York, NY, USA, 1988. Association for Computing Machinery.
6. R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 169–188. Springer, 2011.
7. A. Bienstock, D. Escudero, and A. Polychroniadou. On linear communication complexity for (maximally) fluid mpc. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 263–294, Cham, 2023. Springer Nature Switzerland.
8. G. R. Blakley. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*, pages 313–313. IEEE Computer Society, 1979.
9. E. Boyle, N. Gilboa, Y. Ishai, and A. Nof. Efficient fully secure computation via distributed zero-knowledge proofs. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 244–276. Springer, 2020.
10. V. R. Cadambe, S. A. Jafar, and H. Maleki. Distributed data storage with minimum storage regenerating codes - exact and functional repair are asymptotically equally efficient, 2010.
11. H. Cai, Y. Miao, M. Schwartz, and X. Tang. On optimal locally repairable codes with super-linear length. *IEEE Transactions on Information Theory*, 66(8):4853–4868, 2020.
12. I. Cascudo, B. David, L. Garms, and A. Konring. Yolo yoso: fast and simple encryption and secret sharing in the yoso model. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 651–680. Springer, 2022.
13. I. Cascudo, B. David, O. Shlomovits, and D. Varlakov. Mt. random: Multi-tiered randomness beacons. Cryptology ePrint Archive, Paper 2021/1096, 2021. https://eprint.iacr.org/2021/1096.

14. B. Chen, W. Fang, S.-T. Xia, and F.-W. Fu. Constructions of optimal (r, δ) locally repairable codes via constacyclic codes. *IEEE Transactions on Communications*, 67(8):5253–5263, 2019.
15. B. Chen, W. Fang, S.-T. Xia, J. Hao, and F.-W. Fu. Improved bounds and singleton-optimal constructions of locally repairable codes with minimum distance 5 and 6. *IEEE Transactions on Information Theory*, 67(1):217–231, 2020.
16. K. Chida, D. Genkin, K. Hamada, D. Ikarashi, R. Kikuchi, Y. Lindell, and A. Nof. Fast large-scale honest-majority mpc for malicious adversaries. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 34–64, Cham, 2018. Springer International Publishing.
17. A. R. Choudhuri, A. Goel, M. Green, A. Jain, and G. Kaptchuk. Fluid mpc: Secure multiparty computation with dynamic participants. In *Annual International Cryptology Conference*, pages 94–123. Springer, 2021.
18. R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 316–334. Springer, 2000.
19. R. Cramer, C. Xing, and C. Yuan. On the complexity of arithmetic secret sharing. In *Theory of Cryptography Conference*, pages 444–469. Springer, 2020.
20. A. Dalskov and D. Escudero. Honest majority mpc with abort with minimal online communication. In *International Conference on Cryptology and Information Security in Latin America*, pages 453–472. Springer, 2021.
21. I. Damgård, D. Escudero, and A. Polychroniadou. Phoenix: Secure computation in an unstable network with dropouts and comebacks. *Cryptology ePrint Archive*, 2021.
22. I. Damgård, Y. Ishai, M. Krøigaard, J. B. Nielsen, and A. Smith. Scalable multiparty computation with nearly optimal work and resilience. In *Annual International Cryptology Conference*, pages 241–261. Springer, 2008.
23. I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart. Practical covertly secure mpc for dishonest majority–or: breaking the spdz limits. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 2013.
24. I. Damgård, V. Pastro, N. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.
25. B. David, G. Deligios, A. Goel, Y. Ishai, A. Konring, E. Kushilevitz, C.-D. Liu-Zhang, and V. Narayanan. Perfect mpc over layered graphs. In *Annual International Cryptology Conference*, pages 360–392. Springer, 2023.
26. A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.
27. M. Fitzi, M. Hirt, and U. Maurer. Trading correctness for privacy in unconditional multi-party computation. In *Annual International Cryptology Conference*, pages 121–136. Springer, 1998.
28. J. A. Garay, R. Gennaro, C. Jutla, and T. Rabin. Secure distributed storage and retrieval. *Theoretical Computer Science*, 243(1):363–389, 2000.
29. C. Gentry, S. Halevi, H. Krawczyk, B. Magri, J. B. Nielsen, T. Rabin, and S. Yakoubov. Yoso: You only speak once. In *Annual International Cryptology Conference*, pages 64–93. Springer, 2021.
30. P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Transactions on Information Theory*, 58(11):6925–6934, 2012.
31. S. Goparaju, A. Fazeli, and A. Vardy. Minimum storage regenerating codes for all parameters. *IEEE Transactions on Information Theory*, 63(10):6318–6328, 2017.

32. V. Goyal, H. Li, R. Ostrovsky, A. Polychroniadou, and Y. Song. Atlas: efficient and scalable mpc in the honest majority setting. In *Annual International Cryptology Conference*, pages 244–274. Springer, 2021.
33. V. Goyal, A. Polychroniadou, and Y. Song. Unconditional communication-efficient mpc via hall's marriage theorem. In *Annual International Cryptology Conference*, pages 275–304. Springer, 2021.
34. V. Goyal, Y. Song, and C. Zhu. Guaranteed output delivery comes free in honest majority mpc. In *Annual International Cryptology Conference*, pages 618–646. Springer, 2020.
35. X. Guang, J. Lu, and F. Fu. Repairable threshold secret sharing schemes. *CoRR*, abs/1410.7190, 2014.
36. Y. Guo, R. Pass, and E. Shi. Synchronous, with a chance of partition tolerance. In *Annual International Cryptology Conference*, pages 499–529. Springer, 2019.
37. V. Guruswami, C. Xing, and C. Yuan. How long can optimal locally repairable codes be? *IEEE Transactions on Information Theory*, 65(6):3662–3670, 2019.
38. J. Han and L. A. Lastras-Montano. Reliable memories with subline accesses. In *2007 IEEE International Symposium on Information Theory*, pages 2531–2535, 2007.
39. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In *annual international cryptology conference*, pages 339–352. Springer, 1995.
40. C. Huang, M. Chen, and J. Li. Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems. In *Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)*, pages 79–86, 2007.
41. L. Jin, L. Ma, and C. Xing. Construction of optimal locally repairable codes via automorphism groups of rational function fields. *IEEE Transactions on Information Theory*, 66(1):210–221, 2020.
42. S. Kadhe and A. Sprintson. Security for minimum storage regenerating codes and locally repairable codes. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1028–1032, 2017.
43. M. Keller, E. Orsini, and P. Scholl. Mascot: faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 830–842, 2016.
44. M. Keller, V. Pastro, and D. Rotaru. Overdrive: making spdz great again. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 158–189. Springer, 2018.
45. V. Kher and Y. Kim. Securing distributed storage: Challenges, techniques, and systems. In *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*, StorageSS '05, page 9–25, New York, NY, USA, 2005. Association for Computing Machinery.
46. T. M. Laing and D. R. Stinson. A survey and refinement of repairable threshold schemes. *Journal of Mathematical Cryptology*, 12(1):57–81, 2018.
47. T. Lavaur and J. Lacan. zkbeacon: Proven randomness beacon based on zero-knowledge verifiable computation. In S. D. C. di Vimercati and P. Samarati, editors, *Proceedings of the 19th International Conference on Security and Cryptography, SECRYPT 2022, Lisbon, Portugal, July 11-13, 2022*, pages 406–414. SCITEPRESS, 2022.
48. J. Li, T. Li, and J. Ren. Secure regenerating code. In *2014 IEEE Global Communications Conference*, pages 770–774, 2014.

49. R. Li, S. Yang, Y. Rao, and Q. Fu. On binary locally repairable codes with distance four. *Finite Fields and Their Applications*, 72:101793, 2021.
50. X. Li, L. Ma, and C. Xing. Optimal locally repairable codes via elliptic curves. *IEEE Transactions on Information Theory*, 65(1):108–117, 2019.
51. L. Ma and C. Xing. A survey on optimal locally repairable codes (in chinese). *SCIENTIA SINICA Mathematica*, pages 1–18, 2-21.
52. U. Martínez-Peñas and F. R. Kschischang. Universal and dynamic locally repairable codes with maximal recoverability via sum-rank codes. *IEEE Transactions on Information Theory*, 65(12):7790–7805, 2019.
53. D. S. Papailiopoulos and A. G. Dimakis. Locally repairable codes. *IEEE Transactions on Information Theory*, 60(10):5843–5855, 2014.
54. S. Pawar, S. El Rouayheb, and K. Ramchandran. On secure distributed data storage under repair dynamics. In *2010 IEEE International Symposium on Information Theory*, pages 2543–2547, 2010.
55. R. Rachuri and P. Scholl. Le mans: Dynamic and fluid mpc for dishonest majority. In *Annual International Cryptology Conference*, pages 719–749. Springer, 2022.
56. M. Raikwar and D. Gligoroski. Sok: Decentralized randomness beacon protocols, 2022.
57. K. V. Rashmi, N. B. Shah, and P. V. Kumar. Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction. *IEEE Transactions on Information Theory*, 57(8):5227–5239, 2011.
58. N. Raviv, N. Silberstein, and T. Etzion. Constructions of high-rate minimum storage regenerating codes over small fields. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 61–65, 2016.
59. A. S. Rawat. A note on secure minimum storage regenerating codes. *CoRR*, abs/1608.01732, 2016.
60. A. S. Rawat. Secrecy capacity of minimum storage regenerating codes. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1406–1410, 2017.
61. B. Sasidharan, G. K. Agarwal, and P. V. Kumar. A high-rate msr code with polynomial sub-packetization level. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2051–2055, 2015.
62. N. Saxena, G. Tsudik, and J. H. Yi. Efficient node admission and certificateless secure communication in short-lived manets. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):158–170, 2008.
63. N. B. Shah, K. V. Rashmi, and P. V. Kumar. Information-theoretically secure regenerating codes for distributed storage. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5, 2011.
64. N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff. *IEEE Transactions on Information Theory*, 58(3):1837–1852, 2012.
65. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
66. N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. Optimal locally repairable codes via rank-metric codes. In *2013 IEEE International Symposium on Information Theory*, pages 1819–1823, 2013.
67. D. R. Stinson and R. Wei. Combinatorial repairability for threshold schemes. *Des. Codes Cryptography*, 86(1):195—-210, jan 2018.
68. C. Suh and K. Ramchandran. On the existence of optimal exact-repair mds codes for distributed storage, 2010.

69. I. Tamo and A. Barg. A family of optimal locally recoverable codes. *IEEE Transactions on Information Theory*, 60(8):4661–4676, 2014.
70. R. Tandon, S. Amuru, T. C. Clancy, and R. M. Buehrer. Toward optimal secure distributed storage systems with exact repair. *IEEE Transactions on Information Theory*, 62(6):3477–3492, 2016.
71. Y. Wu, D. Li, and F. Wang. Secret sharing member expansion protocol based on ecc. *The Open Cybernetics & Systemics Journal*, 8(1), 2014.
72. M. Ye and A. Barg. Explicit constructions of optimal-access mds codes with nearly optimal sub-packetization. *IEEE Transactions on Information Theory*, 63(10):6307–6317, 2017.
73. M. Ye, H. Qiu, Y. Wang, Z. Zhou, F. Zheng, and T. Ma. A method of repairing single node failure in the distributed storage system based on the regenerating-code and a hybrid genetic algorithm. *Neurocomputing*, 2020.
74. J. Yu, F. Kong, and R. Hao. Publicly verifiable secret sharing with enrollment ability. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 3, pages 194–199. IEEE, 2007.

Appendix

A Comparison with a Two-Level Shamir's Secret Sharing Scheme

In this section, we will compare our construction presented in Construction 1, which we will denote by Σ with a more natural two-level Shamir's secret sharing scheme, which we denote by Σ'. For completeness, first, we discuss the two-level Shamir's secret sharing scheme Σ'. Let n be the number of parties and consider two integers v and m such that $n = (v+1)m$. Split the parties into m groups of $v+1$ parties each. Let q be a prime number, d, w integers with $d \leq v$ and $w \leq m-1$. Consider a linear secret-sharing scheme that, to distribute a secret $s \in \mathbb{F}_q$, proceeds as follows.

1. Generate m shares of the secret s using a (w,m)–Shamir's secret sharing scheme where the m shares are denoted as $s_1, \ldots, s_m$. Here for positive integers $a < b$, we use the notation (a,b)–Shamir's secret sharing scheme to denote the Shamir's secret sharing scheme providing a privacy where shares are evaluations of a polynomial of degree a in b distinct evaluation points. Suppose that this is done by using a degree w polynomial $F(\mathtt{X}) = \sum_{i=0}^{w} F_i \mathtt{X}^i$ with set of evaluation points $\{\alpha_1, \ldots, \alpha_m\}$.
2. For each $i = 1, \ldots, m$, generate $v+1$ shares for the local secret s_i using a $(d-1, v+1)-$ Shamir's secret sharing scheme where the $v+1$ shares are denoted as $s_{i,1}, \ldots, s_{i,v+1}$. Suppose that such share generation is done using a degree $d-1$ polynomial $F^{(i)}(\mathtt{X}) = \sum_{j=0}^{d-1} F_j^{(i)} \mathtt{X}^j$ with $\{\gamma_{i,1}, \ldots, \gamma_{i,v+1}\}$ as the set of evaluation points.
3. For $i = 1, \ldots, m$ and $j = 1, \ldots, v+1$, assign the share $s_{i,j}$ to the j-th party in group i.

Note that we have in total m local groups, each with $v+1$ players where the threshold for the local group is $d-1$. On the other hand, when we consider each group as one player, our construction is reduced to a (w,m)- Shamir's secret sharing scheme. In order to have a comparable scheme, for the second construction, which is based on two-steps of Shamir's secret sharing, we assume that the first step uses a (w,m)-Shamir's secret sharing scheme where each of its share is further secretly shared using a $(d-1, v+1)$-Shamir's secret sharing scheme. Furthermore, for the first step of the sharing, we assume that the chosen evaluation points are $\alpha_0, \ldots, \alpha_m$ where α_0 is used as the evaluation point for the secret. Furthermore, for $K = 1, \ldots, M$, each such share is further secretly shared using evaluation points $\gamma_1^{(K)}, \ldots, \gamma_{v+1}^{(K)}$. The complete specification of the secret sharing scheme Σ' is specified in Algorithm 6.

Note that in this definition, we assume that the m instantiations of the secret sharing schemes used to secretly share S_K to $v+1$ players using the polynomial $f^{(K)}$ encodes S_K as $f^{(K)}(0)$. However, it is easy to see that a simple shifting operation can be used to have S_K to be $f^{(K)}(y_K)$ for any choice of $y_K \in \mathbb{F}_q$ without changing any of the shares $S_i^{(K)}$.

Algorithm 6 Two-Step Shamir's Secret Sharing Scheme

Require: $S \in \mathbb{F}_q$: the secret to be secretly shared.

1: Randomly select $F_0, \ldots, F_w \in \mathbb{F}_q$ such that $S = \sum_{i=0}^{w} F_i \alpha_0^i$;
2: **for** $K = 1, \ldots, m$ **do**
3: Define $S_K = \sum_{i=0}^{w} F_i \alpha_K^i$;
4: Randomly select $f_0^{(K)}, \ldots, f_{d-1}^{(K)}$ such that $S_K = f_0^{(K)}$;
5: Define $f^{(K)}(\mathtt{X}) = \sum_{j=0}^{d-1} f_j^{(K)} \mathtt{X}^j$;
6: **Secret and shares:** Calculate and distribute the shares to the $v+1$ players where the share for the player assigned $\gamma_i^{(K)}$ is $S_i^{(K)} = f^{(K)}(\gamma_i^{(K)})$;
7: **end for**

First we show that for any secret sharing scheme Σ constructed using Algorithm 1, it is equivalent to Σ', which is obtained from Algorithm 6 with some choice of the parameters.

Lemma 9. *Let $(S_{i,j} : 1 \le i \le m, 1 \le j \le v+1)$ be the shares for the n players in a secret sharing scheme Σ constructed using Algorithm 1 with a fixed secret $S \in \mathbb{F}_q$. Then there exists some assignments of the variables such that the shares generated by a secret sharing scheme Σ' following Algorithm 6 are $(S_{i,j} : 1 \le i \le m, 1 \le j \le v+1)$.*

Proof. Let $\alpha_0 = g(0)$ and for $i = 1, \ldots, m, \alpha_i = g(\beta_i)$. For each $K = 1, \ldots, M$ and $j = 1, \ldots, v+1$, we define $\gamma_j^{(K)} = \gamma_{K,j}$. We aim to show that for any $K = 1, \ldots, M$ and $j = 1, \ldots, v+1$, we have $S_i^{(K)} = S_{K,i}$. For $i = 0, \ldots, w$, we set $F_i = a_{0i}$. Then we have $\sum_{i=0}^{w} F_i \alpha_0^i = \sum_{i=0}^{w} a_{0i} g(0)^i = S$ as required. Furthermore, we have $S_K = \sum_{i=0}^{w} F_i \alpha_K^i = \sum_{i=0}^{w} a_{0i} g(\beta_K)^i$. Lastly, we set $f_0^{(K)} = S_K$ and for $j = 1, \ldots, d-1$ and $K = 1, \ldots, m$, let $f_j^{(K)} = \sum_{i=0}^{w} a_{ji} g(\beta_K)^i$.

Then for each group $K = 1, \ldots, m$, the player assigned to $\gamma_i^{(k)}$ receives the share

$$\begin{aligned} S_i^{(K)} = f^{(K)}(\gamma_i^{(K)}) &= \sum_{j=0}^{d-1} f_j^{(K)} \left(\gamma_i^{(K)}\right)^j \\ &= \sum_{i=0}^{w} a_{0i} g(\beta_K)^i + \sum_{j=1}^{d-1} \left(\sum_{i=0}^{w} a_{ji} g(\beta_K)^i \right) \left(\gamma_i^{(K)}\right)^j \\ &= \sum_{j=0}^{d-1} \sum_{i=0}^{w} a_{ji} g(\gamma_i^{(K)})^j \left(\gamma_i^{(K)}\right)^j = S_{K,i} \end{aligned}$$

completing the proof. □

Next we show that a secret sharing scheme Σ' obtained from Algorithm 6 is also equivalent to a secret sharing scheme Σ obtained from Algorithm 1 with some possible changes of parameters.

Lemma 10. *Let $(S_i^{(K)} : K = 1, \ldots, m, i = 1, \ldots, v+1)$ be the shares for the n players generated by Algorithm 6 with a fixed secret $S \in \mathbb{F}_q$ and the evaluation points $\gamma_i^{(K)}$ that are pairwise distinct. Then there exists some assignments of the variables such that $(S_i^{(K)} : K = 1, \ldots, m, i = 1, \ldots, v+1)$ are generated using Algorithm 1 with some possibly changed parameters.*

Proof. Recall that we have $S = \sum_{i=0}^{w} F_i \alpha_0^i$. Next, for $K = 1, \ldots, m$, we have $f_0^{(K)} = s_K = \sum_{i=0}^{w} F_i \alpha_K^i$. Furthermore, we define $f_1^{(K)}, \ldots, f_{d-1}^{(K)}$ such that $S_i^{(K)} = \sum_{j=0}^{d-1} f_j^{(K)} \left(\gamma_i^{(K)}\right)^j$.

Then for $K = 1, \ldots, m$, we have $f^{(K)}(\mathtt{X}) = \sum_{i=0}^{w} F_i \alpha_K^i + \sum_{j=1}^{d-1} f_j^{(K)} \mathtt{X}^j$. Now for $j = 1, \ldots, d-1$, there exists $d_j \in \{0, \ldots, m-1\}$ and $a_{0,j}, \ldots, a_{d_j,j} \in \mathbb{F}_q$ such that for any $K = 1, \ldots, m$, $f_j^{(K)} = \sum_{i=0}^{d_j} a_{i,j} \alpha_K^i$. We further define $d_0 = w$ and $a_{i,0} = F_i$ for $i = 0, \ldots, w$. Lastly, we define $D_1 = \max\{d_0, \ldots, d_{d-1}\}$. For any $i = 0, \ldots, D_1$ and $j = 1, \ldots, d-1$, such that $a_{i,j}$ is not yet defined, we set $a_{i,j} = 0$.

Then we have $f^{(K)}(\mathtt{X}) = \sum_{j=0}^{d-1} \left(\sum_{i=0}^{D_1} a_{i,j} \alpha_K^i\right) \mathtt{X}^j$ where for any $i = 1, \ldots, v+1$, we have $S_i^{(K)} = f^{(K)}(\gamma_i^{(K)}) = \sum_{j=0}^{d-1} \left(\sum_{i=0}^{D_1} a_{i,j} \alpha_K^i\right) \left(\gamma_i^{(K)}\right)^j$. This shows that if we want to have one polynomial $f^*(\mathtt{X})$ such that for any $K = 1, \ldots, m$ and $i = 1, \ldots, v+1$, $f^*(\gamma_i^{(K)}) = S_i^{(K)}$, we need to have some polynomial $g^*(\mathtt{X})$ such that for any $K = 1, \ldots, m$ and $i = 1, \ldots, v+1$, $g^*(\gamma_i^{(K)}) = \alpha_K$. Note that such $g^*(\mathtt{X})$ is guaranteed to exist with degree of at most n. Then with such $g^*(\mathtt{X})$, we can define $f^*(\mathtt{X}) = \sum_{j=0}^{d-1} \left(\sum_{i=0}^{D_1} a_{i,j} g * (\mathtt{X})^i\right) x^j$. So letting $\gamma_{i,j} = \gamma_j^{(i)}$ for $i = 1, \ldots, M, j = 1, \ldots, v+1$, for such specific instances of the secret sharing schemes obtained by Algorithm 6 it can also be generated using Algorithm 1. □

Remark 3. We note that based on the form of f^* which requires the polynomial $g^*(x)$, for two players belonging to different local groups, they cannot possess the same evaluation points. Otherwise, they will have exactly the same share. So this is the reason why we require such restriction in the statement of Lemma 10.

Lemma 10 shows that if all the evaluation points in the second step of Algorithm 6 are pairwise distinct, then we can transform it to a one-step secret sharing which follows Algorithm 1 with a possible change in the degree of $g(\mathtt{X})$ from $v+1$ to a positive integer, say D_2, and the inner degree from w to D_1. We claim that $v+1 \leq D_2$. Indeed, we note that $g^*(\mathtt{X})$ cannot be a constant since we have at least m distinct evaluation points evaluated to m distinct values. Furthermore, recall that $g^*(\gamma_i^{(1)}) = \alpha_1$ for $i = 1, \ldots, v+1$. Consider $\hat{g}(\mathtt{X}) = g^*(\mathtt{X}) - \alpha_1$. It is easy to see that $\hat{g}(\mathtt{X})$ has the same degree as $g^*(\mathtt{X})$ and neither is a constant function. However, we have $\hat{g}(\gamma_i^{(1)}) = 0$ for $i = 1, \ldots, v+1$. Hence we have $\prod_{i=1}^{v+1} (\mathtt{X} - \gamma_i^{(1)}) | \hat{g}(\mathtt{X})$, proving that its degree, and hence the degree of $g^*(\mathtt{X})$, is at least $v+1$ as claimed.

Consider Σ a secret sharing scheme following Algorithm 1. By Theorems 1 and 2, Σ is shown to have t_1 privacy and r_1 recovery where $(d-1)(w+1) \leq t_1 \leq d(w+1)-1$ and $r_1 \leq w(v+1)+d$.

Now suppose that we generate a secret sharing scheme Σ' following Algorithm 6. First we consider its recovery level r_2. Note that by Lemma 10, the degree of the constructed $f^*(\mathtt{X})$ is $D_1 \cdot \deg(g^*(\mathtt{X})) + d - 1$ where $\deg(g^*(\mathtt{X}))$ is the degree of $g^*(\mathtt{X})$. Recall that $D_1 = \max\{d_0, d_1, \ldots, d_{d-1}\}$ where $d_0 = w$ while for any $j = 1, \ldots, d-1$, the polynomial $\sum_{i=0}^{d_j} a_{i,j}\mathtt{X}^i$ maps α_K^i to $f_j^{(K)}$ for each $K = 1, \ldots, m$. Note that since $f_j^{(K)}$ is also unknown, each of such d_j can be as large as $m-1$. Hence, in general, $d_j = m-1$ for $j = 1, \ldots, d-1$, which implies $D_1 = m-1$. Furthermore, as we have established, the degree of $g^*(\mathtt{X})$ is at least $v+1$, which implies that $\deg(f^*(\mathtt{X})) \geq (m-1)(v+1)+(d-1) = m(v+1)-(v+2-d)$.

Consider a group of players containing $v+1$ players from the first w groups and $d-1$ players from the remaining $m-w$ groups. Note that such group has size $(v+1)w + (d-1)(m-w) \leq \deg(f^*(\mathtt{X}))$. We claim that it is possible to have a share generation such that the original secret $s = 1$ while all the shares of these players to be 0. Note that by linearity of Σ', this means that the shares from such group of size $m(v+1) + (m-w)(d-v-2)$ contains no information about s, proving that it provides privacy from such group. This would imply that $r_2 \geq (v+1)w + (d-1)(m-w) + 1$.

First, for the first w groups, since we have the shares of all $v+1$ players from such group, we would be able to recover the local secret s_i from the shares of these players. Since their shares are 0, we can conclude that $s_i = 0$ for $i = 1, \ldots, w$. Next, note that since s_i is a valid share from a (w, m)-Shamir's secret sharing scheme, by the w-privacy guarantee of the (w, m)-Shamir's secret sharing scheme, there is a valid share generation of 1 such that $s_i = 0$ for $i = 1, \ldots, w$. This will also fix the values of s_i for $i = w+1, \ldots, m$. Next, we consider the sharing of the players in the i-th group for $i = w+1, \ldots, m$. Note that since $s_{i,j}$ is a valid secret share of s_i using $(d-1, v+1)$-Shamir's secret sharing scheme, by its $d-1$-privacy guarantee, it is possible to have a valid share generation of s_i such that $s_{i,j} = 0$ for $d-1$ of such j. This shows that it is possible to have a valid share generation of $s = 1$ such that the share of all the players belonging to the group described above to be zero. This proves that $r_2 \geq (v+1)w + (d-1)(m-w) + 1$. Combined with the upper bound established for r_1, we obtain $r_2 - r_1 \geq (d-1)(m-w-1)$.

Next, we consider the privacy level of Σ', which we denote by t_2. Consider the group of players consisting of d players from each of the first $w+1$ groups. It is easy to see that such group can recover the original secret. This shows that $t_2 \leq d(w+1)-1$. So combined with the fact that $(d-1)(w+1) \leq t_1$, we have $t_2 - t_1 \leq d-1$.

It is easy to see that when $w - 1 < m$, the increase of recovery level, which is at least $(m-w-1)(d-1)$ is at least the increase in the privacy level, which is at most $d-1$. This gap becomes much larger especially in the scenario where the secret sharing scheme is (strongly) multiplicative with repairing process that provides statistical security for the privacy of the shares. Note that in this case, since $w \leq \frac{m}{2}$ and $m = O\left(\frac{n}{v}\right) = O\left(\frac{n}{\ln^{(1+\varepsilon)} n}\right)$ for some $\varepsilon > 0$, the gap is

$O\left(\frac{n}{\ln^{(1+\varepsilon)} n}\right)$. This shows that in general, a secret sharing scheme generated by Algorithm 6 comes with a larger gap between the privacy level and recovery level. So if we maintain the recovery level to be the same, the privacy level provided by Σ' is much smaller than what can be guaranteed from the construction following Algorithm 1. Such limitation of Algorithm 6 provides us with a justification on considering the one-step construction in Algorithm 1 instead of the more natural Algorithm 6.