On the Cost of Reconstructing a Secret, or VSS with Optimal Reconstruction Phase

Ronald Cramer, Ivan Damgård, and Serge Fehr*

Aarhus University, BRICS
{cramer,ivan,fehr}@brics.dk

Abstract. Consider a scenario where an l-bit secret has been distributed among n players by an honest dealer using some secret sharing scheme. Then, if all players behave honestly, the secret can be reconstructed in one round with zero error probability, and by broadcasting nl bits.
We ask the following question: how close to this ideal can we get if up to t players (but not the dealer) are corrupted by an adaptive, active adversary with unbounded computing power? - and where in addition we of course require that the adversary does not learn the secret ahead of reconstruction time. It is easy to see that $t = \lfloor (n-1)/2 \rfloor$ is the maximal value of t that can be tolerated, and furthermore, we show that the best we can hope for is a one-round reconstruction protocol where every honest player outputs the correct secret or "failure". For any such protocol with failure probability at most $2^{-\Omega(k)}$, we show a lower bound of $\Omega(nl + kn^2)$ bits on the information communicated. We further show that this is tight up to a constant factor.
The lower bound trivially applies as well to VSS schemes, where also the dealer may be corrupt. Using generic methods, the scheme establishing the upper bound can be turned into a VSS with efficient reconstruction. However, the distribution phase becomes very inefficient. Closing this gap, we present a new VSS protocol where the distribution complexity matches that of the previously best known VSS, but where the reconstruction phase meets our lower bound up to a constant factor. The reconstruction is a factor of n better than previous VSS protocols. We show an application of this to multi-party computation with pre-processing, improving the complexity of earlier similar protocols by a factor of n.

1 Introduction

The concept of *secret-sharing* (introduced by Shamir [13]) is of fundamental importance: in practical data security, as a way to protect a secret simultaneously from exposure and from being lost; and theoretically, as the basis for building general multi-party secure protocols.

In the original setting of Shamir, a *dealer* distributes a secret, say an l-bit string, to n players, by privately sending a *share* to each player. The computation

* This work was done while the 3$^{\text{rd}}$ author was employed at ETH Zürich, supported by the Swiss National Science Foundation, project no. SPP 2000-055466.98.

of the shares is done w.r.t. a threshold value t, where $1 \leq t \leq n$. Later, some subset of the players can attempt to reconstruct the secret by pooling their shares. A secret sharing scheme must ensure *privacy*, i.e., an adversary who sees up to t of the shares learns no information about the secret, and *correctness*, i.e., the secret can always be reconstructed from a set of at least $t+1$ shares.

Here, we will first consider a more adversarial setting where up to t of the players (but not the dealer) may be corrupted by an active, adaptive and unbounded adversary, in particular, corrupted players may contribute incorrect shares (or nothing) in the reconstruction phase. We still require privacy, and also correctness in the sense that the honest players can reconstruct the correct secret. Consider the following question. How much information must be sent in order for such a scheme to work? This question is interesting only if $n/3 \leq t < n/2$, since otherwise the problem is either "too hard" or "too easy": if $t \geq n/2$ the problem clearly cannot be solved, and if $t < n/3$, standard methods (see [2]) immediately give an optimal solution with zero error probability.

Somewhat surprisingly, little work seems to have been done on the case of $n/3 \leq t < n/2$ (although upper bounds follow from known protocols [12, 4]). It is easy to see that for t in this range, one cannot construct a scheme where the correct secret is *always* reconstructed. At best one can make a scheme where every honest player outputs the correct secret or "failure", where the latter happens with probability only $2^{-\Omega(k)}$, where k is a security parameter. For schemes that achieve this for the maximal value of t, i.e. $t = \lfloor (n-1)/2 \rfloor$, and where the reconstruction is completed in a single round, we show a lower bound of $\Omega(nl + kn^2)$ bits on the amount of information sent in the reconstruction. This may be seen as an answer to the question "what does it cost to get the best possible security in a minimal number of rounds?". No such bound was known previously, and it holds even for schemes that are not efficient.

We refer to the type of scheme we just described as *Honest-Dealer VSS*. This is because the well-known concept of Verifiable Secret Sharing (VSS), introduced in [6], is essentially what we just described, except that also the dealer can be corrupt. In VSS, distributing the secret may then take the form of an interactive, several rounds protocol. One usually assumes that a private channel connects every pair of players and that a broadcast channel is available[1]. A secure VSS must, in addition to what we required above, also ensure that immediately after the distribution phase, some value of the secret is uniquely defined (even if the dealer is corrupt) and that this value will be reconstructed (with overwhelming probability). Note that the standard definition of VSS is slightly weaker than ours in that it allows honest players to reconstruct (with small probability) an incorrect value of the secret, even if the adversary was passive in the distribution phase. However, all known VSS protocols for our communication model (see e.g. [12, 4]) satisfy or can trivially be modified to satisfy our stronger definition.

Our lower bound for Honest-Dealer VSS trivially applies also to VSS (we cannot expect to do better in a more adversarial situation).

[1] The latter can be simulated by the private ones if $t < n/3$, but must be assumed as a separate primitive otherwise.

For an honest dealer, we use known results on authentication codes to show that the lower bound is tight up to a constant factor (even if we count the total information sent). This scheme establishing the upper bound is computationally efficient and can - at least in principle - be turned into a VSS, since the honest dealer could always be replaced by a secure multi-party computation using generic methods (e.g. [12, 4]). This, however, is not a satisfactory solution: while reconstruction would be the same complexity as before, the distribution would become extremely inefficient in comparison. To close this gap, we present a new VSS protocol where the complexity of the distribution matches that of the previously best known VSS for our scenario [4], but where the reconstruction meets our lower bound. This beats previous VSS protocols by a factor of n.

We show an application of this to multi-party computation with pre-processing, introduced in [1], where the n players ultimately want to compute a function f on private inputs $x_1, \ldots, x_n$. In order to do this more efficiently than starting from scratch, the players are allowed to a pre-processing and store some information obtained in this phase *before* the function and the inputs become known. The computation phase of our protocol has communication complexity $O(n^2k|C|)$, where $|C|$ is the size of the circuit to be computed. This improves the computation phase of earlier similar protocols by a factor of n without increasing the complexity of the pre-processing.

In the appendix, we sketch how our results for a dishonest minority generalize for almost all t in the range $n/3 \leq t < n/2$ and observe that already an arbitrarily small linear gap between t and $n/2$ allows to reduce the communication complexity of the reconstruction by a factor of n. Using methods from [5], we also show how to generalize our schemes to provide security against any (non-threshold) Q^2 adversary (see [9]), improving known results by a factor of at least n. Finally, we look at the case where the reconstruction is allowed to use more than one round of interaction and observe, using results from [7], that the amount of information sent by the honest dealer can be brought down to $n(n+k)$ bits, at the expense of a significantly more inefficient reconstruction phase.

2 Communication Model

Throughout the paper, we consider the *secure-channels model with broadcast* [12], i.e. there is a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of n players plus a so called *dealer* D, every two entities being connected by a secure, untappable channel, and there is a broadcast channel available. We assume an *active* adversary with unbounded computing power that can corrupt up to a certain number t out of the n players in $\mathcal{P}$ plus the dealer D. An adversary is *rushing*, if he can learn the messages sent by the honest players in each round before deciding on the messages for corrupted players in this round. Finally, the adversary can either be *static* or *adaptive*, the former meaning that he has to corrupt the players before the protocol execution and the latter that he can corrupt players at his will during the protocol execution, depending on what he has seen so far. Throughout the paper, we consider a *security parameter* k.

3 Single-Round Honest-Dealer VSS

We first model the general communication pattern for VSS schemes where the dealer is guaranteed to be honest and whose reconstruction phase consists of a single round of communication. We will call such a scheme *Single-Round Honest-Dealer VSS*. Our main point of interest is the communication complexity of the reconstruction phase of such a scheme. Consider schemes of the following general form, and assume an active adversary who corrupts up to t of the n players P_i, *but not the dealer* (this is also known as *robust secret sharing*).

Distribution Phase: The honest dealer generates shares $s_i = (k_i, y_i)$, $i = 1 \dots n$, according to a fixed and publicly known conditional probability distribution $P_{S_1 \cdots S_n|S}(\cdots|s)$, where s is the secret. Privately he sends s_i to player P_i.

Reconstruction Phase: Each player P_i is required to broadcast $\tilde{y}_i$, which is supposedly equal to y_i. Locally and by some fixed (possibly probabilistic) method, each player P_i decides on the secret s based on his private k_i and on the broadcast $\tilde{y}_1, \dots, \tilde{y}_n$, i.e., either outputs a value $\tilde{s}$, hopefully equal to s, or outputs "failure".

It is not difficult to see that in fact we may always and without loss of generality assume our schemes of interest to be of this form (please refer to Appendix A).

For each of the at most t corrupted players P_j, the adversary can broadcast a manipulated $\tilde{y}_j$, which may depend arbitrarily on the private information $s_j = (k_j, y_j)$ of those corrupted players, or broadcast nothing at all in some cases ("crash faults"). Note though that for at least $n - t$ $\tilde{y}_i$'s it holds that $\tilde{y}_i = y_i$. If additionally the adversary is rushing, he can choose to "speak last" in the reconstruction phase. This means that in principle any corrupted shares may additionally depend on the information broadcast by the honest players, in particular they may depend on the secret s. By contrast, a non-rushing adversary is one who selects the corrupted shares before the start of the reconstruction phase. Note that security against non-rushing adversaries makes sense in a communication model enhanced with a "simultaneous broadcast channel", i.e., one by means of which all players broadcast their information at the same time.

We define our notion of security. Assume an active adversary that corrupts at most t of the n players but not the dealer. Additionally, the adversary can be static or adaptive, and rushing or non-rushing. A Single-Round Honest-Dealer VSS scheme is $(t, n, 1-\delta)$-*secure* if the following holds.

Privacy: As a result of the distribution phase, the adversary gains no information about the secret s distributed by the honest dealer.

$(1-\delta)$-*Correctness*: In the reconstruction phase, each uncorrupted player outputs either the correct secret s or "failure", where for every player the latter happens with probability at most $\delta < 1$, independent of s.
In the special case that the adversary introduces only crash-faults or remains passive, all honest players recover the correct secret s with probability 1.

As mentioned in the Introduction, we focus on the case of a *dishonest minority*, i.e., $t = \lfloor (n-1)/2 \rfloor$, the maximal value of t for which $(t, n, 1-\delta)$-security is achievable. For the corresponding results for a (nearly) arbitrary t in the range $n/3 \leq t < n/2$, we refer to Appendix C. Note that the case $t < n/3$ is completely understood: zero failure probability and optimally efficient communication can be achieved by a combination of Shamir's secret sharing scheme and standard efficient error correction techniques [2].

We stress that our definition of security captures the best one can achieve in this setting. Negligible error δ^m is achieved by m parallel repetitions. More importantly, it only differs from perfect security in the sense that there is a (small) probability that some player does not reconstruct the secret and outputs "failure" instead. This is unavoidable in the presence of an arbitrary (not necessarily rushing) active adversary, as is easy to see (please refer to Appendix B). Furthermore, existing Honest-Dealer VSS schemes like [12] ("secret sharing when the dealer is a knight") fulfill our security definition without any changes in the required communication.

A seemingly stronger security definition would require agreement among the honest players in all cases, i.e., they all recover the correct secret or they all output "failure", where the latter would happen with probability at most δ. However, this is impossible to achieve in a single round reconstruction phase with a *rushing adversary*, as we show in Appendix B. [2]

Note also that the reconstruction procedure in our definition is completely general in that it does not dictate how the correct secret is recovered by the honest players. The definition merely states that from all broadcast and from his private information, an honest player can reconstruct the secret. In particular, in our definition it need not be the case that an honest player, using his private information, "filters out" false shares and reconstructs the secret from the "good" ones, as it is the case for known schemes [12, 4] and the one we present later.

4 Lower Bound on Reconstruction Complexity

We prove the following lower bound. Note that the standard definitions of entropy, conditional entropy, mutual information and conditional mutual information are used throughout this section. We refer to [3] for an excellent introduction to information theory.

Theorem 1. *For any family of Single-Round Honest-Dealer VSS schemes, $(t, n, 1-\delta)$-secure against an active, rushing adversary, the following holds. If $t = \lfloor (n-1)/2 \rfloor$ and $\delta \in 2^{-\Omega(k)}$ for a security parameter k, then the total information broadcast in the reconstruction phase is lower bounded by $\Omega(nH(S) + kn^2)$.*

Note that it is immaterial whether the adversary is adaptive or not.

In the following, we will call K_i the *key* and Y_i the *public share* of player P_i. Theorem 1 follows immediately from

[2] In Appendix E, we argue that agreement is possible in the presence of a non-rushing adversary. Agreement can be achieved in all cases by adding one extra round of communication.

Proposition 1. *Let $S_1 = (K_1, Y_1), \ldots, S_n = (K_n, Y_n)$ be distributed according to the Single-Round Honest-Dealer VSS scheme. Then, in case of an odd n, the size of any public share Y_i is lower bounded by*

$$H(Y_i) \in \Omega(H(S) + kn),$$

while for an even n, it is the size $H(Y_i Y_j)$ of every pair $Y_i \neq Y_j$ that is lower bounded by $\Omega(H(S) + kn)$.

We will only prove the case of an odd n, i.e., $n = 2t + 1$; the proof for an even n, i.e. $n = 2t + 2$, goes accordingly. But before going into the proof, consider the following Lemma, which states a well known result from Authentication Theory, which can be found in various literature starting with [14] (for a very general treatment of Authentication Theory consult [11]).

Lemma 1. *Let K, M, Y and Z be random variables (typically key, message, tag and public information of an authentication scheme) with joint distribution P_{KMYZ} such that M is independent of K and Z but uniquely defined by Y and Z. Then, knowing Z, one can compute $\tilde{Y}$, consistent with K and Z with probability*

$$p_I \geq 2^{-I(K;Y|Z)}.$$

Also, knowing Z and Y, one can compute $\tilde{\tilde{Y}}$, consistent with K and Z and a $\tilde{\tilde{M}} \neq M$ with probability

$$p_S \geq 2^{-H(K|Z)}.$$

In the context of Authentication Theory, $\tilde{Y}$ describes an *impersonation* and $\tilde{\tilde{Y}}$ a *substitution* attack, and p_I and p_S are the corresponding success probabilities.

In the proof of Proposition 1, we apply the following Corollary, which follows from the fact that a successful impersonation attack is also a successful substitution attack with probability at least $1/2$, assumed that M is uniformly distributed among a set of cardinality at least two.

Corollary 1. *Let K, M, Y and Z be as above, except that M is required to be uniformly distributed among a non-trivial set. Then, knowing Z, one can compute $\tilde{Y}$, consistent with K and Z and a $\tilde{M} \neq M$ with probability*

$$p_S \geq 2^{-I(K;Y|Z)-1}.$$

Proof of Proposition 1: Since by the privacy of the scheme the public share Y_i is independent of S and hence $H(Y_i)$ does not depend on the distribution of S, we can assume P_S to be the uniform distribution. Furthermore, for symmetry reasons, we can focus on the public share of the player P_{t+1}.
Let $i \in \{1, \ldots, t\}$ be arbitrary but fixed, and consider an adversary corrupting the first $i - 1$ players $P_1, \ldots, P_{i-1}$ as well the player P_{t+1}. One of the goals of the adversary could be to substitute P_{t+1}'s public share Y_{t+1} by a false share $\tilde{Y}_{t+1}$ that is consistent with the public shares $Y_1, \ldots, Y_t$ of the first t

players and player P_i's key K_i (and maybe even the keys $K_1, \ldots, K_{i-1}$), but that leads to an incorrect secret $\tilde{S} \neq S$. Indeed, if the adversary succeeds in this attack, from player P_i's point of view, the $t+1$ public shares $Y_1, \ldots, Y_t, \tilde{Y}_{t+1}$ *could* come from honest and the t shares $Y_{t+2}, \ldots, Y_n$ from corrupted players. Hence, P_i clearly cannot compute the correct secret with certainty, and so outputs "failure". Therefore, the success probability of this attack is at most $\delta \in 2^{-\Omega(k)}$. On the other hand however, according to the above Corollary, applied to $K = K_i$, $M = S$, $Y = Y_{t+1}$ and $Z = (K_1, \ldots, K_{i-1}, Y_1, \ldots, Y_t)$, the success probability is at least $p_S \geq 2^{-I(K_i; Y_{t+1} | K_1 \cdots K_{i-1} Y_1 \cdots Y_t) - 1}$. Therefore, we have $I(K_i; Y_{t+1} | K_1 \cdots K_{i-1} Y_1 \cdots Y_t) \in \Omega(k)$. This holds for every $i \in \{1, \ldots, t\}$, and hence, using the chain rule for mutual information, we get

$$I(K_1 \cdots K_t; Y_{t+1} | Y_1 \cdots Y_t) = \sum_{i=1}^{t} I(K_i; Y_{t+1} | Y_1 \cdots Y_t\, K_1 \cdots K_{i-1}) \in \Omega(kt)$$

and therefore $H(Y_{t+1}) \geq I(K_1 \cdots K_t; Y_{t+1} | Y_1 \cdots Y_t) \in \Omega(kt) = \Omega(kn)$.

As $S_1, \ldots, S_t$ gives no information about S, but $S_1, \ldots, S_t, Y_{t+1}$ determines S, we also have $H(Y_{t+1}) \geq H(S)$, and hence $H(Y_{t+1}) \in \Omega(H(S) + kn)$. □

In Appendix E we illustrate the power of rushing by giving an example of a concrete scheme secure against a non-rushing adversary, that beats the lower bound, and sketch a tight lower bound result. We also briefly discuss the minimal complexity of the distribution phase of schemes secure against a rushing adversary.

5 Tightness of the Lower Bound

We first describe a very natural, generic construction of a Single-Round Honest-Dealer VSS and then present a particular instantiation that meets the lower bound from the previous section. Rabin and Ben-Or [12] first considered a solution of this type. The scheme below differs from theirs only in the choice of the authentication code (which, however, will be relevant later on).

Let a $(t+1, n)$-threshold secret-sharing scheme be given as well as an authentication scheme, e.g. based on a family of strongly universal hash functions $\{h_\kappa\}_{\kappa \in \mathcal{K}}$ (see e.g. [15]). To share a secret s, the dealer D generates shares $s_1, \ldots, s_n$ according to the secret sharing scheme, and, for each pair of players P_i, P_j, he selects a random authentication key $\kappa_{ij} \in \mathcal{K}$ which will be sent to P_j who will later use it to verify a share contributed by P_i. Then D computes for each share s_i and for each P_j the authentication tag $y_{ij} = h_{\kappa_{ij}}(s_i)$ that should be revealed by P_i at reconstruction time to convince P_j that P_i's share s_i is valid. D then simply sends shares, tags and keys privately to the players who own them. To reconstruct, every player broadcasts his share together with the tags (or, alternatively, sends to every player his share and the corresponding tag), and verifies the authenticity of the received shares using his keys.

We use Shamir's secret sharing scheme [13] over a field F with $|F| > n$, and the well-known family of hash functions $h_{(\alpha,\beta)}(X) = \alpha X + \beta$ defined over F. The success probability of a substitution attack of the corresponding authentication scheme is $1/|F|$. It follows that the probability of player P_i accepting a false share from another player is $1/|F|$, and hence the probability of player P_i not reconstructing the correct secret is at most $t/|F|$. By comparing all the accepted shares with the reconstructed sharing polynomial and outputting "failure" in case of inconsistencies, he makes sure not to output an incorrect secret. Hence, choosing F such that $|F|$ is in $2^{\Theta(k)}$ (assuming n to be at most polynomial in k), we have the following upper bound, already achieved in [12].

Theorem 2. *For $t = \lfloor (n-1)/2 \rfloor$, there exists a Single-Round Honest-Dealer VSS scheme, $(t, n, 1-2^{-\Omega(k)})$-secure against an adaptive and rushing adversary, with a total communication complexity of $O(kn^2)$ bits.*

A remark concerning the authentication code. The choice of the code is not completely arbitrary, since it is important for our later purposes that computation of tags has low arithmetic complexity (here one multiplication and one addition over F) and that the tags are *linear* if α is fixed, as shown in Section 7.1.

6 Upper Bound in the Presence of a Corrupted Dealer

In this section, we present a VSS scheme with a one-round reconstruction, where the complexity of the distribution phase matches that of the previous best known VSS for our scenario [4], but where the reconstruction phase meets our lower bound up to a constant factor. This is at least a factor of n better than previous VSS protocols.

6.1 Definition

Since now the dealer might be corrupt as well and so the distribution of the secret takes the form of an interactive protocol, the adversary can not only intrude faults in the reconstruction, but also in the distribution. Therefore, our definition operates with two error probabilities, which for a concrete scheme do not have to be equal: first the probability that the distribution fails to work as supposed, and second the probability that the reconstruction fails, even though the distribution succeeded.

Assume an active adversary that corrupts at most t of the n players plus the dealer (respectively, including the dealer, in case he is one of the players). Additionally, the adversary can be static or adaptive, and rushing or non-rushing. Consider a scheme with an arbitrary distribution phase resulting in every player P_i holding a key k_i and a public share y_i and with a one-round reconstruction phase as in the honest dealer case. We call such a scheme $(t, n, 1-\beta, 1-\delta)$-*secure* if, except with probability β (taken over the coin flips during the distribution), the following holds.

Privacy: As long as the dealer remains honest, the adversary gains no information about the shared secret s as a result of the distribution phase.

$(1-\delta)$-*Correctness*: Once all currently uncorrupted players complete the distribution phase, there exists a fixed value s' such that in the reconstruction phase each uncorrupted player outputs either s' or "failure", where for every player the latter happens with probability at most $\delta < 1$, independent of s'. If the dealer remains uncorrupted during the distribution, then $s' = s$.
In the special case that the adversary introduces only crash-faults or remains passive, all honest players recover s' with probability 1.

Again, existing VSS schemes essentially fulfill our stronger definition, in particular the most efficient solution known, [4], fulfills it without any changes in the required communication, while the [12] protocol requires some straightforward modifications.

6.2 Towards VSS with Optimized Reconstruction

The security of the scheme from the last section evidently completely breaks down in case the dealer is corrupted. In the distribution phase, he could hand out inconsistent shares and inconsistent authentication tags, and, in the reconstruction phase, since he knows all the keys, he could compute correct tags for false shares. This would allow him to disrupt the reconstruction and even to actually cause different secrets to be reconstructed (see the analysis in [4] of WSS from [12]). To remedy this, we have to ensure that the players that remain honest receive consistent shares, and that they accept each others shares at reconstruction, while rejecting false shares. Of course, as mentioned in the introduction, this could in principal be achieved by replacing the dealer of the Honest-Dealer VSS by a general MPC. This, however, would result in a rather inefficient distribution phase. Also the following approach seems to be no satisfactory solution because of the same reason. We force the dealer to distribute consistent shares $s_1, \ldots, s_n$ by doing a "two-dimensional sharing" as in [2] or [4] and then every tag y_{ij} for a share s_i is computed in a multi-party fashion, such that it is guaranteed to be correct and the corresponding key is only known to the verifier P_j. Again, doing general MPC would result in a rather inefficient distribution phase; however, the following points provide some intuition as to why the full generality of MPC protocols is not needed, and instead we can do a *specialized* MPC.

1. A "two-dimensional sharing" from [2] or [4] not only ensures that the uncorrupted players hold consistent shares, but also that every share s_i is again correctly shared. Hence, one input to the MPC, s_i, is already correctly shared.
2. We only have to guarantee that a tag is computed correctly, if the player who will later verify it is honest at distribution time. At reconstruction, a corrupted player can always claim a tag to be invalid, even if it were good. For this reason, full VSS of the authentication key will not be necessary.
3. The function to be computed uses only one multiplication and one addition. This will allow us to do the distributed multiplication locally, i.e. no resharing as in [8] will be needed.

6.3 The CDDHR VSS Sharing Protocol

To describe the sharing protocol from [4], we start by reviewing the concept of Information Checking (IC), introduced in [12]. In essence, an IC scheme provides unconditionally secure "signatures" with limited transferability. More concretely, it allows a *sender* S to provide a *transmitter* T (also called *intermediary*) with a message m and a "signature" σ, such that T can later pass (m, σ) on to a *recipient* R, claiming that m originates with S. The signature σ enables R to verify this. We use the notation $\sigma_m(S,T;R)$ to refer to such a signature. Although in reality the "signing" procedure is an interactive protocol involving all three players and using a broadcast channel, we abuse language slightly and simply say that S "sends the signature $\sigma_m(S,T;R)$ to T". IC must fulfill the following requirements, except with some small error probability. If T and R are uncorrupted, then R indeed accepts T's message m (*consistency*). If, on the other hand, S and R are uncorrupted, then R rejects any message $m' \neq m$ (*correctness*). Finally, if S and T are uncorrupted, then R gets no information on m before T passes (m, σ) on to him (*secrecy*). It is easy to extend this concept and the corresponding protocols to multiple recipients, say $R_1, \ldots, R_n$, by simply executing the single recipient protocol for each possible recipient. We then use the notation $\sigma_m(S,T) = (\sigma_m(S,T;R_1), \ldots, \sigma_m(S,T;R_n))$. For a formal definition and technical details, please refer to [12, 4].

Please recall that the IC-signatures from [4] over a field F have the following *linearity* properties. If T holds two signatures $\sigma_m(S,T;R)$ and $\sigma_{m'}(S,T;R)$ and if λ is known to R and T, then T can compute a signature $\sigma_{m+m'}(S,T;R)$ for $m + m'$ and a signature $\sigma_{\lambda m}(S,T;R)$ for λm. This holds analogously in the multi-recipient case. As to efficiency, generating a signature $\sigma_m(S,T;R)$ costs $O(\log |F|)$ bits of communication, generating a signature $\sigma_m(S,T)$ with n recipients costs $O(n \log |F|)$ bits of communication. Furthermore, the secrecy condition holds perfectly while correctness and consistency hold with probability $1 - 2^{-\log |F|}$ for a single-recipient and $1 - 2^{-\log |F| + \log(n)}$ for a multi-recipient signature.

We present the VSS sharing protocol from [4], which we will call $\mathsf{PreShare}$, in a slightly modified version. Namely, for ease of exposition, we use a *symmetrical* polynomial and we omit the signatures made by the dealer (since these are needed only to catch a corrupted dealer early on).

Protocol $\mathsf{PreShare}$

1. To share a secret $s \in F$, the dealer chooses a random symmetrical bivariate polynomial f of degree at most t in both variables with s as constant coefficient, i.e. $f(0,0) = s$.
2. To every player P_i, the dealer privately sends the *actual share* $s_i = f(i,0)$ and the sharing $s_{i1} = f(i,1), \ldots, s_{in} = f(i,n)$ of s_i. [3]

[3] In the descriptions of all the protocols, whenever a player expects to receive a message from another player, but no message arrives or it is not in the right format, he takes some fixed default value as received message.

3. For every two players P_i and P_j, the following is done. P_i sends s_{ij} together with a signature $\sigma_{s_{ij}}(P_i, P_j) = (\sigma_{s_{ij}}(P_i, P_j; P_1), \ldots, \sigma_{s_{ij}}(P_i, P_j; P_n))$ to P_j. If $s_{ij} \neq s_{ji}$, then P_j broadcasts a complaint, to which the dealer has to answer by broadcasting s_{ji}. If this value does not coincide with P_j's s_{ji}, then P_j accuses the dealer publicly who then has to broadcast P_j's share s_j and subshares $s_{j1}, \ldots, s_{jn}$. [4]
4. If at some point, the broadcast information is inconsistent, the players take some publicly known default sharing.

This protocol stands as a VSS sharing protocol on its own (but with "expensive" reconstruction, as argued earlier). The proof of this fact is based on the following observations. Please refer to [4] or the appendix.

Proposition 2. *After the execution of* PreShare*, every honest P_i holds a share s_i and signed sub-shares $s_{i1} \ldots s_{in}$ such that*

1. *If the dealer remains honest, then the adversary has no information about the secret s.*
2. *The sub-shares $s_{i1} \ldots s_{in}$ of any honest player P_i are a correct sharing of s_i, and $s_{ij} = s_{ji}$ holds for all P_i and P_j who remain honest.*
3. *The shares s_i of the honest players are correct shares of a unique value s', which is the secret s if the dealer remains honest.*
4. *For any (honest or dishonest) player P_j, the sub-shares s_{ij} of the honest players P_i are correct shares of P_j's share s_j, which is well defined by the shares s_i of the honest players.*

The communication complexity of this PreShare protocol is $O(n^3 \log |F|)$ bits, the dealer essentially distributes n^2 sub-shares and each of these sub-shares is signed, where signing costs $O(n \log |F|)$ bits of communication per signature.

6.4 Computing Tags by a Specialized MPC

Consider now a fixed player P_i after the execution of PreShare, holding his share s_i and the corresponding sub-shares $s_{i1}, \ldots, s_{in}$ with signatures $\sigma_{s_{i1}}(P_1, P_i), \ldots, \sigma_{s_{in}}(P_n, P_i)$. We now want to compute authentication tags $y_{ij} = \alpha_{ij} \cdot s_i + \beta_{ij}$ for s_i as they are computed by the dealer in the Honest-Dealer VSS protocol, but without letting the dealer know the keys, $(\alpha_{ij}, \beta_{ij})$ should only be known to P_j.

At the heart, there is the following problem. A player P wants to compute the tag $y = \alpha \cdot m + \beta$ for his secret message m with respect to a player V's secret key α, β. As already mentioned earlier, this will be done by a *specialised* MPC.

We assume that P's message m is already correctly shared by shares $m_1, \ldots, m_n$ and that P holds signatures $\sigma_{m_1}(P_1, P; V), \ldots, \sigma_{m_n}(P_n, P; V)$, verifiable by V. If the protocol PreShare from the previous section has been executed, and if P's message m stands for P_i's share s_i, then this is fulfilled with $m_k = s_{ik}$ and $\sigma_{m_k}(P_k, P; V) = \sigma_{s_{ik}}(P_k, P_i; P_j)$.

[4] Of course, broadcast values do not have to be signed anymore; however, for simpler notation, we assume that also broadcast sub-shares s_{ij} are signed by $\sigma_{s_{ij}}(P_i, P_j)$.

Protocol MP Auth

1. V chooses a random polynomial f_α of degree at most t with $f_\alpha(0) = \alpha$ and a random polynomial f_β of degree at most $2t$ with $f_\beta(0) = \beta$. For every player P_k, V sends the shares $\alpha_k = f_\alpha(k)$ and $\beta_k = f_\beta(k)$ to P_k together with signatures $\sigma_{\alpha_k}(V, P_k; P)$ and $\sigma_{\beta_k}(V, P_k; P)$, verifiable by P.
2. Every player P_k, having received the shares α_k and β_k with the corresponding signatures and holding the share m_k of m, computes $y_k = \alpha_k \cdot m_k + \beta_k$ and, using the linearity property of the signatures, the corresponding signature $\sigma_{y_k}(V, P_k; P)$ [5] and passes y_k and $\sigma_{y_k}(V, P_k; P)$ on to P, who verifies the signature (see point 3. in Section 6.2).
3. If P receives all the y_k and all the signatures are good, then he can reconstruct y by interpolation, i.e. by computing a polynomial f_y of degree at most $2t$ with $f_y(k) = y_k$ for all P_k and computing $y = f_y(0)$.
   If some signature $\sigma_{y_k}(V, P_k; P)$ is not correct, then before computing y as above, P passes m_k and $\sigma_{m_k}(P_k, P; V)$ on to V, who verifies the signature and in case of a good signature returns $y_k = \alpha_k \cdot m_k + \beta_k$ to P (see point 2. in Section 6.2 for the case V refuses).

Proposition 3. *Under the assumptions stated before the protocol, the following holds except with probability* $2^{-\log|F|+O(\log n)}$.

1. *If P and V remain honest during the execution, then $y = \alpha \cdot m + \beta$.*
2. *If P remains honest, then the adversary learns nothing about m.*
3. *If V remains honest, then the adversary learns nothing about α.*

Hence, the tag y can be thought of being computed by some honest player.

Proof. We will prove 1., 2. and 3. under the assumption that the security properties of the signatures hold without error probability; this proves the claim.

1. Let f_m be the polynomial of degree at most t with $f_m(k) = m_k$ and hence $f_m(0) = m$. The n shares $y_k = \alpha_k \cdot m_k + \beta_k$ define a unique polynomial f_y of degree at most $2t$ with $f_y(k) = y_k$ and $f_y(0) = y = \alpha \cdot m + \beta$, namely $f_y = f_\alpha \cdot f_m + f_\beta$. So, if all n players P_k behave and send y_k with the correct signature to P, then P can compute f_y and hence y. If on the other hand some corrupted player P_k misbehaves and sends an incorrect y_k to P (or an incorrect signature or nothing at all), then P recognizes this and gets the correct y_k from V. Hence, even in this case P gets all the correct y_k and can therefore reconstruct y.
2. We assume wlog that V is corrupted. If all the corrupted players P_k follow the protocol, then the adversary definitely gets no information at all. If some corrupted player P_k misbehaves (e.g. by sending a bad y_k), then the adversary only learns m_k, which he already knows.
3. We assume that P is corrupted. Note that the adversary does not learn anything new by asking V for a y_k in step 3., since the correct value m_k must be sent to V (otherwise V would not accept the signature and return nothing).

[5] Note that m_k is known to both P_k and P.

We have to show that the adversary's view of this protocol gives no information about α. The adversary's view, excluding the signatures, consists of m, $m_1, \ldots, m_n$, $y_1, \ldots, y_n$ and α_k and β_k for $P_k \in A$, where A is the set of corrupted players, with $y_k = \alpha_k \cdot m_k + \beta_k$. Consider the polynomial $d_\alpha(X) = \prod_{P_k \in A} (k-X)/k$ of degree t and the polynomial $d_\beta = -d_\alpha \cdot f_m$ of degree at most $2t$. Note that $d_\alpha(0) = 1$ and $d_\beta(0) = -m$ and $d_\alpha(k) = 0 = d_\beta(k)$ for all P_k in A. This implies that if f_α and f_β are the sharing polynomials for α and β, then for any α', β' with $\alpha' \cdot m + \beta' = y$, the polynomials $f_{\alpha'} = f_\alpha + (\alpha' - \alpha)d_\alpha$ and $f_{\beta'} = f_\beta + (\alpha' - \alpha)d_\beta$ are sharing polynomials for α' and β', consistent with the adversary's view. Note that $f_{\beta'}(0) = \beta - (\alpha' - \alpha)m = y - \alpha' \cdot m = \beta'$. Since f_α and f_β are randomly chosen with $f_\alpha(0) = \alpha$ and $f_\beta(0) = \beta$, the adversary's view of the protocol, excluded the signatures, is independent of α. This together with the secrecy property of the signatures proves the claim. □

The communication complexity of one execution of MP Auth is $O(n \log |F|)$ bits. Namely, V essentially shares α and β. Note that the signatures involved are signatures verifiable by one player, hence they only cost $O(\log |F|)$ bits of communication.

6.5 The VSS Protocol

The VSS sharing protocol that meets the lower bound of Theorem 1 now works as follows. First, Pre Share is applied to the secret and then, by applying MP Auth to the shares, the sub-shares and signatures are stripped off and replaced by tags for the actual shares:

Protocol Share

1. The above protocol Pre Share is executed on the secret s. As a result, every player P_i holds a share s_i, sub-shares $s_{i1}, \ldots, s_{in}$ and signatures $\sigma_{s_{i1}}(P_1, P_i)$, $\ldots, \sigma_{s_{in}}(P_n, P_i)$.
2. For every player P_i, tags $y_{i1}, \ldots, y_{in}$ for s_i are computed by executing MP Auth with every player P_j on the message s_i and P_j's randomly chosen key $(\alpha_{ij}, \beta_{ij})$.

Note that all the sub-shares s_{ij} and signatures $\sigma_{s_{ij}}(P_j, P_i)$ are only temporarily used and can be deleted at the end of the protocol. For the reconstruction, as in the honest-dealer case, only the shares, the tags and the keys are needed.

Theorem 3. *For $t = \lfloor (n-1)/2 \rfloor$, there exists a Verifiable Secret Sharing scheme, $(t, n, 1 - 2^{-\Omega(k)}, 1 - 2^{-\Omega(k)})$-secure against an adaptive and rushing adversary, with a sharing complexity of $O(kn^3)$ and a single-round reconstruction of complexity $O(kn^2)$.*

Proof sketch: We can take the above scheme over a field F with $|F| \in 2^{\Theta(k)}$. Secrecy and correctness follow from Propositions 2 and 3. The communication complexity of the Pre Share protocol is $O(kn^3)$, of the MP Auth protocol it is

$O(kn)$. Therefore, the communication complexity of the sharing protocol, which calls $\mathsf{Pre\,Share}$ once and $\mathsf{MP\,Auth}$ n^2-times, is $O(kn^3)$. The communication complexity of the reconstruction is as in the Honest-Dealer VSS $O(kn^2)$ bits. □

7 Applications to MPC with Pre-processing

As an application of the above described VSS scheme, we will now present a general MPC protocol in the pre-processing model [1]. Our protocol is secure against an active, adaptive adversary who can corrupt up to $t = \lfloor (n-1)/2 \rfloor$, a minority, of the players. The idea behind MPC with pre-processing, introduced by Beaver [1], is to do as much work as possible in a *pre-processing phase*, before the inputs and even the circuit [6] are known. This is to reduce the work and the assumptions on the communication network required in the *computation phase* when the inputs and circuit have actually become available.

This is based on circuit randomization and a generic construction that can be applied to any general MPC protocol based on a VSS with certain linearity properties explained below. The computation phase doesn't require secure channels, it only consists of broadcasting information and performing the local computations necessary for VSS reconstructions. It should therefore be clear that MPC in the pre-processing model benefits from VSS with optimized reconstruction.

The required linearity properties are as follows. If s and s' are two VSS'ed secrets and λ a public constant, then the players should be able to locally compute VSS shares of $s + s'$ and $\lambda \cdot s$ (if this is the case then the scheme is called *homomorphic*) and of $s + \lambda$. Before showing that our VSS has these properties, we sketch the protocol for general MPC with pre-processing. Assume that adequate upperbounds on the number of inputs and multiplication gates in the future circuit are known. In the pre-processing phase, each player chooses a sufficient number of independent random values a and VSS'es them. Next, the players jointly prepare a sufficient number of random triples r, r' and r'' such that $r'' = rr'$ and such that each of these values is VSS'ed. Note that mutual randomness is easily achieved by having players VSS random values, and taking the sum of those as a mutually random value. By the linearity property, this random value is effectively VSS'ed. By invoking the general MPC protocol, products can be securely computed with the result VSS'ed.
In the computation phase, inputs and circuit are known. Assume for simplicity that each player has a single private input value. Each player then takes his actual private input s, and simply broadcasts the difference $\epsilon = a - s$ between this input s and the random value a he VSS'ed in the pre-processing phase. Subsequently, all players locally compute their shares in s from the shares in a they hold and the now public value ϵ. In the computation phase, the addition gates are handled locally while to multiply two shared values s and s', a fresh precomputed random triple (r, r', r'') is taken, the differences $\delta = s - r$ and $\delta' = s' - r'$ are revealed by invoking the reconstruction of VSS. Since

[6] Usually, the function that is to be securely computed is given as an arithmetic circuit

$ss' = (r+\delta)(r'+\delta') = rr' + \delta' r + \delta r' + \delta\delta' = r'' + \delta' r + \delta r' + \delta\delta'$, every player P_i can locally compute a share of ss' from the shares of r, r' and r'' and the values δ and δ'. Note that linearity of the VSS facilitates all of these steps.

7.1 Applying Our VSS to MPC with Pre-processing

We first argue that our VSS can be made to have the required linearity properties. Note that Shamir shares trivially possess these properties, so it suffices to focus on the authentication code. As mentioned in Section 5, the only thing we need to do is to fix throughout the computation the values α that are part of the verification keys (α, β). Indeed, if y and y' are authentication tags for m and m' with keys (α, β) and (α, β'), respectively, then for every $\lambda \in F$, $\lambda \cdot y + y'$ is an authentication tag for the message $\lambda \cdot m + m'$ with key $(\alpha, \lambda \cdot \beta + \beta')$. Namely, $\alpha \cdot (\lambda \cdot m + m') + (\lambda \cdot \beta + \beta') = \lambda \cdot (\alpha \cdot m + \beta) + (\alpha \cdot m' + \beta') = \lambda \cdot y + y'$. Analogue, it can be shown that y is an authentication tag for the message $m + \lambda$ with key $(\alpha, \beta - \alpha \cdot \lambda)$. Furthermore, it is not difficult to see by induction that after l authentications and verifications with the same α, the substitution probability still is $l/(|F| - l + 1)$ (see e.g. [4]).

For a field F with $|F| \in 2^{\Theta(k)}$, the protocol now works as follows. In the pre-processing phase, the random input values a are treated just as above, based on our VSS. In order to prepare the random triples, we use the general MPC techniques of [4] to prepare triples r, r' and r'' with $r'' = rr'$ as described earlier. This results in a VSS of these values according to [4] (i.e., according to the protocol $\mathsf{PreShare}$ from Section 6.3). We can convert these to sharings as they would have been produced by our VSS, we simply apply the protocol MPAuth (see Section 6) to get shares according to Share. Hence, all necessary pre-processing information will be shared according to our VSS. The computation phase can now proceed based on the reconstruction phase of our VSS.

As to efficiency, generating the sharings of r and r' consists essentially of $O(n)$ executions of $\mathsf{PreShare}$, and thus this has complexity $O(kn^3)$ bits. The computation of the sharing of r'' costs according to [4] $O(kn^4)$ bits of communication, assuming everyone coorperates. Multi-party computing the tags is negligible compared to the rest, namely $O(kn^3)$. Hence, we have a best case complexity of $O(kn^4)$. If a corrupted player refuses to coorperate, then the easiest thing to do is to exclude the player and restart the computation. This will allow the adversary to slow down the computation by at most a factor linear in n. [7] Hence we have

Theorem 4. *Let C be an arithmetic circuit over a field F with M multiplication gates, where $|F| \in 2^{\Theta(k)}$. Communicating $O(Mkn^5)$ bits in a pre-processing phase, there exists a MPC protocol, secure, except with probability $2^{-\Omega(k)+M}$, against a rushing adversary who can adaptively corrupt up to $t = \lfloor (n-1)/2 \rfloor$ of the players, computing the circuit C with $O(Mkn^2)$ bits of comunication.*

[7] Instead of restarting, one could also reconstruct the share(s) of the caught cheater, if needed. This way, the adversary cannot slow down the computation substantially, resulting in a pre-processing complexity of $O(Mkn^4)$ instead of $O(Mkn^5)$.

The most efficient previously known protocol for MPC with pre-processing in our model is based on [4]. Note that this would result in a pre-processing phase with complexity of the same order as in our case. However, due to VSS with optimized reconstruction, we gain an efficiency improvement of a multiplicative factor n in the computation phase of our protocol.

References

1. D. Beaver. Efficient multiparty protocols using circuit randomization. In *CRYPTO '91*, LNCS 576, pages 420–432. Springer-Verlag, 1992.
2. M. Ben-Or, S. Goldwasser, and A. Widgerson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th Annual ACM Symposium on the Theory of Computing*, pages 1–10, 1988.
3. R.E. Blahut. *Priciples and Practice of Information Theory.* Addison-Wesley, 1987.
4. R. Cramer, I. Damgard, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multi-party computations secure against an adaptive adversary. In *EUROCRYPT '99*, LNCS 1592. Springer-Verlag, 1999.
5. R. Cramer, I. Damgaard, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT 2000*, LNCS 1807. Springer-Verlag, 2000.
6. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 383–395, 1985.
7. S. Cabello, C. Padró, and G. Sáez. Secret sharing schemes with detection of cheaters for a general access structure. In *Proceedings of the 12th International Symposium on Fundamentals of Computation Theory, FCT '99*, LNCS 1233, pages 185–193, 1999.
8. R. Gennaro, M.O. Rabin, and T. Rabin. Simplified VSS and fast-track multi-party computations with applications to threshold cryptography. In *17th ACM Symposium on Principles of Distributed Computing*, 1998.
9. M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In *16th ACM Symposium on Principles of Distributed Computing*, pages 25–34, 1997.
10. M. Karchmer and A. Wigderson. On span programs. In *8th Annual Conference on Structure in Complexity Theory (SCTC '93)*, pages 102–111, 1993.
11. U. Maurer. Authentication theory and hypothesis testing. *IEEE Transaction on Information Theory*, 2000.
12. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *21th Annual ACM Symposium on the Theory of Computing*, pages 73–85, 1989.
13. A. Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, 1979.
14. G.J. Simmons. Authentication theory/coding theory. In *CRYPTO '84*, LNCS 196, pages 411–431. Springer-Verlag, 1985.
15. D.R. Stinson. *Cryptography — Theory and Practice.* Number ISBN 0-8493-8521-0. CRC Press, 1995.

A Communication Pattern from Section 3: Justification

When justifying the claim that the proposed communication pattern is most general, it should be kept in mind that we are interested in the complexity of the reconstruction phase, and that all "re-modeling" operations are allowed as long as they do not affect the complexity of reconstruction (apart from constant factors).

By the assumption that the dealer is honest, we may assume without loss of generality that the distribution phase only consists of the dealer sending private information s_i to each of the players P_i, i.e., any secure distributed computation carried out by the players in the distribution phase could as well be carried out by the honest dealer, without consequences to the complexity of the reconstruction phase. Similarly, we may assume that in the reconstruction phase each player P_i merely broadcasts a piece of information, y_i, that only depends on the private information s_i received from the dealer. Namely, at the cost of at most a constant factor of increased communication, private channels can be simulated by one-time pads, the keys of which are distributed by the honest dealer. In fact, it can be assumed that in general $s_i = (k_i, y_i)$, where y_i is required to be broadcast in the reconstruction phase, and each player P_i makes a local (possibly probabilistic) decision on the secret s based on the broadcast information and his private k_i.

B Impossibility Lemmas from Section 3

Lemma 2. *There exists a static, non-rushing adversary such that with non-zero probability some honest players output "failure" in the reconstruction phase.*

Proof. Given that $t \geq n/3$, let B, A_0, A_1 be an arbitrary disjoint partition of $\{1, \ldots, n\}$ such that $|B| = t$ and $1 \leq |A_0|, |A_1| \leq t$. We show a strategy for the adversary that forces all players in B to output "failure" with non-zero probability. The adversary corrupts the players in A_0, selects a random secret $\tilde{s}$ and randomly guesses the shares $s_i = (k_i, y_i)$ held by the players in B. By the privacy of the scheme and assuming that he guessed the shares correctly and that $s \neq \tilde{s}$ (which both happens with non-zero probability), he can sample random shares $\tilde{s}_j$ for the corrupted players, so that these, together with the shares of the players in B, are consistent with the secret $\tilde{s}$, and have the same distribution as when sent by the honest dealer. It is now clear that in the reconstruction phase (assumed that the adversary guessed the shares correctly and that $s \neq \tilde{s}$), every player in B has to output "failure". Indeed, the players in B must definitely not output the incorrect secret $\tilde{s}$. On the other hand, if some player in B outputs the correct secret s (with positive probability), then by corrupting the players in A_1 instead of A_0, but otherwise playing the corresponding game, the adversary creates the same view for the players in B, however with the correct and the incorrect secrets exchanged, and hence this player would now output the incorrect secret (with positive probability), which is a contradiction. $\square$

Lemma 3. *There exists a static, rushing adversary such that such that with non-zero probability some honest player recovers the secret in the reconstruction phase, while some other honest player outputs "failure".*

Proof. Consider the case $t > n/3$. Let B, A_0, A_1 be an arbitrary disjoint partition of $\{1, \ldots, n\}$ such that $1 \leq |A_0| \leq t-1$ and $1 \leq |A_1| \leq t$. Note that $2 \leq |B| \leq t$. Let p, q be distinct members of B. We consider the same adversary as before, except that in the reconstruction phase, the adversary "rushes", and waits until the players in B have broadcast their y_i's. He then makes a guess for player p's private k_p, and broadcasts random $\tilde{y}_j$'s for the players, consistent with k_p and with the y_i's of the players in B and a random secret different from the correct one (which he knows by now). For similar reasons as before we conclude that player p does not reconstruct the secret if the guess for k_p was correct. However, in that case player q must reconstruct the secret with positive probability: for if not, corrupting A_0 *and* player p (note that this amounts to at most t corruptions), the adversary would not have to guess k_p anymore, and hence there would be a strategy that makes at least one honest player output "failure" in the reconstruction with probability equal to 1. This contradicts correctness. □

C Non-maximal t

In the main body of this paper, we have only considered a maximal t in the interesting range $n/3 \leq t < n/2$. We will now state the generalizations of the Theorems 1 to 4 for a (nearly) arbitrary t in this range. The corresponding proofs are similar but technically more involved.

Theorem 1'. *For any family of Single-Round Honest-Dealer VSS schemes, $(t, n, 1-\delta)$-secure against an arbitrary active, rushing adversary, the following holds. Let k be a security parameter and let $\epsilon > 0$ be an arbitrary constant. If $\delta = 2^{-\Omega(k)}$ and $n/3 \cdot (1+\epsilon) \leq t < n/2$ then the total information broadcast in the reconstruction phase is lower bounded by $\Omega((nH(S) + kn^2)/(n-2t))$.*

Note that already an arbitrarily small linear gap between t and $n/2$ reduces the lower bound by a factor of n. The following Theorem shows that the reconstruction complexity indeed reduces by a factor of n for such a t (at least in case of a security parameter k slightly larger than linear in n).

Theorem 2'. *For $n/3 \leq t < n/2$ and $k = \Omega((n-2t)\log(t))$, there exists a Single-Round Honest-Dealer VSS scheme, $(t, n, 1-2^{-\Omega(k)})$-secure against an adaptive and rushing adversary, with a total communication complexity of $O(kn^2/(n-2t))$ bits.*

The according holds for VSS.

Theorem 3'. *For $n/3 \leq t < n/2$ and $k = \Omega((n-2t)\log(t))$, there exists a Single-Round VSS scheme, $(t, n, 1-2^{-\Omega(k)}, 1-2^{-\Omega(k)})$-secure against an adaptive and rushing adversary, with a sharing complexity of $O(kn^3)$ and a reconstruction complexity of $O(kn^2/(n-2t))$.*

Applied to MPC with preprocessing, we achieve

Theorem 4'. *Let C be an arithmetic circuit over a field F with M multiplication gates, where $|F| \in 2^{\Theta(k)/(n-2t)}$, $n/3 \leq t < n/2$ and $k = \Omega((n-2t)\log(t))$. Communicating $O(Mkn^5)$ bits in a pre-processing phase, there exists a MPC protocol, secure, except with probability $2^{-\Omega(k)+M}$, against an adversary who can adaptively corrupt up to t of the players, computing the circuit C with $O(Mkn^2/(n-2t))$ bits of communication.*

D General Adversaries

We now go beyond security against a dishonest *minority* by sketching how to adjust our VSS and MPC protocols to be secure against a *general Q^2-adversary* [9], i.e. against an adversary who can corrupt any subset of players in a given family of subsets, where no two subsets in the family cover the full player set.

By replacing the bivariate polynomial sharing in Pre Share by the information-theoretic commitment/WSS protocol from [5] based on monotone span programs [10], we are in the same position as described by Proposition 2, except that 4. is not guaranteed, i.e. the share s_i of a corrupted player P_i is not necessarily correctly shared by the sub-shares s_{ji} of the honest players P_j. But this can easily be achieved by doing another level of sharing: every player P_i shares his share s_i with the WSS protocol from [5] where every player P_j insists that the share they get of s_i is the sub-share s_{ji}.
In the MP Auth protocol, replacing the threshold sharings of the values α and β by sharings based on monotone span program sharings [10] with multiplication, and using the fact that these can be constructed from ordinary monotone span programs with only constant overhead [5], Proposition 3 remains intact.
This results in a VSS scheme secure against a general Q^2-adversary. Furthermore, the sharing and reconstruction complexities are $O(knm^2)$ and $O(knm)$ bits, respectively, where $m \geq n$ is the size of the monotone span program, while the respective complexities of the general adversary VSS scheme suggested in [4] are both $O(knm^3)$ bits (even though one could achieve $O(knm^2)$ using their techniques in a more elaborate way).

Based on this general adversary VSS scheme, similar to the previous section, one can achieve a general MPC protocol, secure against a general Q^2 adversary, which in the pre-processing model has a communication complexity of $O(Mknm)$ bits, compared to $O(Mknm^3)$ (respectively $O(Mknm^2)$), which would be achieved by the general adversary MPC protocol from [4].

E The Power of Rushing (Honest Dealer Case)

We show that our tight lower bound from Section 4 does not hold if the adversary does not rush, and instead selects the corrupted shares he will broadcast in the reconstruction phase before it has started. We also sketch a lower bound and outline some applications, namely to a scenario in which the amount of information sent in the *distribution phase* is to be minimized.

Let F be a finite field with $|F| > n$, and take Shamir's $(t+1, n)$-threshold scheme defined over F. Cabello, Padró and Sáez [7] have proposed the following so-called robust secret sharing scheme. To share a secret s in this scheme, the honest dealer selects a random field element ρ, independently generates full sets of Shamir-shares for the secrets s, ρ and $\rho \cdot s$, and privately distributes the shares to the players.

Given a set A of at least $t+1$ shares (which possibly contains corrupted shares), consider the three values s', ρ' and τ' that are computed by applying the reconstruction procedure of Shamir's scheme to the shares in A. The crucial observation is that if $s \neq s'$ and if the corrupted shares are *independently distributed from* ρ, the probability that $s' \cdot \rho' = \tau'$ is at most $1/|F|$. Hence, given for instance a trusted party available for reconstruction, connected with each player by an independent private channel, the independence requirement is satisfied and although the secret may not always be reconstructed from a qualified set, a corrupted secret is detected with high probability.

We note the following application of this scheme in our scenario of a non-rushing adversary. By assumption, this is an adversary who chooses the corrupted shares before the reconstruction phase. This ensures that the independence requirement stated before is satisfied. Let k be a security parameter and $t < n/2$ and assume additionally that $|F| \geq 2^{n+k}$. Let the distribution phase be according to the scheme of [7]. Consider an arbitrary set A of $t+1$ shares revealed in the reconstruction phase. If A consists exclusively of shares of honest players, then the secret s_A reconstructed by the procedure above would certainly be the correct secret s. Else, either a failure would be detected, or with probability at most 2^{-n-k}, a secret $s_A \neq s$ is accepted based on the shares in A. Let V denote the set of all distinct *accepted* "secrets" s_A, by quantifying over all sets A. Note that $s \in A$. Now each honest player simply computes V, and outputs "failure" if V has more than one element, and s otherwise. This way all honest players are in agreement, and the probability with which they output "failure" is clearly at most $2^n \cdot 2^{-n-k} = 2^{-k}$.

For the case that $k > n$ and $n = 2t+1$, we now sketch an argument showing that the distribution phase of this scheme is optimal, up to constant factors. A basic result in secret sharing says, informally speaking, that the size of individual shares is at least the size of the secret, and hence the question that remains is whether the error probability ϵ of the above scheme is optimal. We define an adversary who flips a random coin and either corrupts the first t players, or the last t players. In either case, he makes a random guess $\tilde{S}_{t+1}$ for the share S_{t+1} that player P_{t+1} received from the honest dealer in the distribution phase, deletes the correct shares received from the dealer by the corrupted players, and instead chooses random corrupted shares, consistent with his guess $\tilde{S}_{t+1}$ and with a random secret $\tilde{s}$. Assuming that the correct secret s was chosen at random by the honest dealer, if the adversaries' guess for player P_{t+1}'s share is correct, then there is no way for any reconstruction procedure to distinguish between s and

$\tilde{s}$. Hence, in order for $\log 1/\epsilon$ to be $O(k)$, the size of each individual share must be $\Omega(k)$.

Although it is generally not very realistic to assume that the adversary is not rushing, it is possible to construct a "simultaneous broadcast" channel on top of the "secure channels with broadcast model". Namely, simply have each player first VSS their values, e.g. by using the schemes of [12, 4], after which all VSS's are opened. Using the concrete scheme above, this procedure would ensure that shares are "broadcast simultaneously", and hence that the required independence is achieved, at the cost of increased complexity of the reconstruction phase and use of private channels in that phase. The advantage, however, is that the efficiency of the *distribution phase* has been substantially improved.

F Proof of Proposition 2

1. First note that the adversary does not gain any new information by making players complain. Let A be the set of players who have been corrupted during the execution of $\mathsf{Pre\,Share}$. The existence the symmetrical polynomial
$$d(X, Y) = \prod_{P_i \in A} \frac{(X-i)(Y-i)}{i^2}$$
of degree t, with $d(0,0) = 1$ and $d(i, \cdot) = d(\cdot, i) = 0$ for all $P_i \in A$, implies that for every $s' \in F$, the number of bivariate symmetrical polynomials of degree at most t with s' as constant coefficient and consistent with the adversary's view is the same. Therefore, as f is chosen at random, the shares and sub-shares of the corrupted players give no information about the secret s. The claim now follows from the secrecy property of the signatures.
2. If this was not the case, then there would have been complaining.
3. Let the set A consist of $t+1$ honest players. Their shares define a unique secret s'. Let now A' consist of the players in A and a further honest player (if there are only $t+1$ honest players, then we are finished anyway). Let λ_i, $i \in A$, be the reconstruction coefficients for the players in A and λ'_i, $i \in A'$, for the players in A'. So we have $s' = \sum_{i \in A} \lambda_i s_i$ and (according to 2.) $s_k = \sum_{i \in A} \lambda_i s_{ki} = \sum_{j \in A'} \lambda'_j s_{kj}$ for all $k \in A'$. It follows that $\sum_{k \in A'} \lambda'_k s_k = \sum_{k \in A'} \lambda'_k \sum_{i \in A} \lambda_i s_{ki} = \sum_{i \in A} \lambda_i \sum_{k \in A'} \lambda'_k s_{ki} = \sum_{i \in A} \lambda_i \sum_{k \in A'} \lambda'_k s_{ik} = \sum_{i \in A} \lambda_i s_i = s'$, hence the shares of the players in A' are still consistent. Inductively, it follows that the shares of all honest players are consistent and define a unique secret s'.
4. Can be shown with a similar argumentation as above using the fact that every share s_j can be written as a fix linear combination $\sum_k \mu_k s_k$ of the shares of the honest players P_k. $\square$