J. Cryptol. (2010) 23: 580–593
DOI: 10.1007/s00145-010-9056-z

Journal of
CRYPTOLOGY

On d-Multiplicative Secret Sharing*

Omer Barkol†
HP Labs, Haifa, Israel
omer.barkol@hp.com

Yuval Ishai
Computer Science Department, Technion, Haifa, Israel
yuvali@cs.technion.ac.il
and
UCLA, Los Angeles, CA, USA

Enav Weinreb‡
VERIX, Glil-Yam, Israel
weinreb@cs.technion.ac.il

Communicated by Ronald Cramer

Received 22 October 2008 and revised 22 December 2009
Online publication 5 February 2010

Abstract. A *multiplicative* secret sharing scheme allows players to multiply two secret-shared field elements by locally converting their shares of the two secrets into an additive sharing of their product. Multiplicative secret sharing serves as a central building block in protocols for secure multiparty computation (MPC). Motivated by open problems in the area of MPC, we introduce the more general notion of *d-multiplicative* secret sharing, allowing to locally multiply d shared secrets, and study the type of access structures for which such secret sharing schemes exist.

While it is easy to show that d-multiplicative schemes exist if no d unauthorized sets of players cover the whole set of players, the converse direction is less obvious for $d \geq 3$. Our main result is a proof of this converse direction, namely that d-multiplicative schemes do not exist if the set of players is covered by d unauthorized sets. In particular, t-private d-multiplicative secret sharing among k players is possible if and *only if* $k > dt$.

Our negative result holds for arbitrary (possibly inefficient or even *nonlinear*) secret sharing schemes and implies a limitation on the usefulness of secret sharing in the context of MPC. Its proof relies on a quantitative argument inspired by communication complexity lower bounds.

Key words. Secret sharing, Secure multiparty computation, Secure multiplication.

* Research supported by grant 1310/06 from the Israel Science Foundation and grant 2004361 from the U.S.-Israel Binational Science Foundation. Work done in part while the authors were visiting the Institute for Pure & Applied Mathematics (IPAM) at UCLA.

† Work done in part at the Computer Science Department, Technion.

‡ Work done in part at the Computer Science Department, Technion, and at CWI Amsterdam.

1. Introduction

Secret sharing schemes allow a dealer to share a secret among a set of players, such that only some pre-defined authorized subsets of the players are able to reconstruct the secret from their shares. The notion of secret sharing was introduced by Shamir [27] and Blakely [7], independently, for the threshold case where the authorized sets are those whose cardinality is larger than a given threshold. Later, Ito et al. [22] generalized this notion to a setting where the authorized subsets are an arbitrary[1] family of subsets of the players; this family of authorized subsets is called an *access structure*. Secret sharing schemes are used in many cryptographic and distributed computing applications, such as Byzantine agreement and distributed coin-flipping, distributed storage, threshold cryptography, private information retrieval and locally decodable error-correcting codes. Most notably, secret sharing is employed as a central building block in protocols for secure multiparty computation (MPC) [6,9,15]. This latter application of secret sharing serves as the primary motivation for the current work.

The application of secret sharing in the context of unconditionally secure MPC protocols relies on an additional *multiplication* property, allowing players to multiply two secret-shared field elements by locally converting their shares of the two secrets into an additive sharing of their product. More precisely, in a multiplicative secret sharing scheme each player P_i, given its share a_i of a secret field element a and its share b_i of a secret field element b, can locally compute a field element c_i such that the outputs c_i add up to ab. The multiplication property of Shamir's threshold scheme [27] was implicitly used in the early MPC protocols of [6,9]. The general notion of multiplication for linear secret sharing schemes[2] was first introduced and studied by Cramer, Damgård, and Maurer [13] and has since received a considerable amount of attention. In particular, efficient constructions of multiplicative secret sharing schemes were recently given in [10,11,25]. The study of multiplicative secret sharing is motivated not only by the natural application to secure *multi-party* computation, but also by recent applications to zero-knowledge proofs and secure two-party computation [16,20,21].

It is known that multiplicative secret sharing is possible if and only if the access structure is of type Q_2, namely there are no two unauthorized sets whose union covers the entire set of players [13]. The negative result can be proved via a reduction to the two-party case and by using known impossibility results for unconditionally secure two party computation [17,23]. Furthermore, Cramer et al. [13] obtained strong positive results about the efficiency of multiplicative secret sharing. Specifically, any linear secret sharing scheme can be transformed into a multiplicative linear secret sharing scheme for the same access structure while only doubling the original share size. A stronger notion of multiplication, termed *strong multiplication*, was also introduced in [13] and motivated by its application to perfectly secure MPC in the presence of an active adversary. Strong multiplication requires that even if an arbitrary unauthorized set of players

[1] Here and in the following, an access structure is required to be *monotone*, in the sense that if A is an authorized set and $A \subseteq A'$ then A' is authorized as well.

[2] In a *linear* secret sharing scheme, the shares are obtained by applying a linear function to the secret and random field elements picked by the dealer. In this paper, we consider the multiplication property in general (possibly nonlinear) secret sharing schemes. We note, however, that the multiplication property implies some limited form of linearity, namely that the shares of a secret can be *locally converted* into additive shares of the same secret.

is excluded, the scheme still retains its (standard) multiplication property. In contrast to standard multiplication, the efficiency of strongly multiplicative schemes realizing general access structures is far less understood and remains an intriguing open question. However, the class of access structures for which secret sharing with strong multiplication is possible is well understood: A necessary and sufficient condition is that the access structure be of type Q_3, namely no union of *three* unauthorized sets covers the entire set of players. Necessity follows by an easy reduction to the case of standard multiplication.

In this paper, we consider a different natural extension of the basic multiplication property of secret sharing that we call *d-multiplication*. The d-multiplication property generalizes standard multiplication by considering a multiplication of d (rather than two) secrets. Specifically, a secret sharing with d-multiplication allows to multiply d secret-shared field elements by enabling the players to locally convert shares of d different secrets into an additive sharing of their product. In contrast to the case of strong multiplication, even the question of characterizing the class of adversary structures that admit a (possibly inefficient or even nonlinear) scheme with d-multiplication does not seem straightforward when $d \geq 3$.

One direction is easy: both Shamir's scheme (in the threshold[3] case) and the replication-based secret scheme of Ito et al. [22] (in the general case) are d-multiplicative if the access structure is of type Q_d. In particular, t-private d-multiplicative secret sharing among k players is possible if $k > dt$. (By t-private we mean that every set of t players is unauthorized, without making any further requirement on the authorized sets.) The main focus of the current work is on whether the above positive result is optimal. In particular:

> Is there any $d \geq 3$ for which there exists a 1-private d-multiplicative secret sharing scheme involving d or fewer players?

We note that in the case of *linear* secret sharing, a negative answer follows from a fairly simple linear algebra argument; see Corollary 1 in [30]. However, such an argument does not apply to general secret sharing. Moreover, in contrast to the cases of multiplication and strong multiplication discussed above, here one cannot resort to standard impossibility results for unconditionally secure MPC. Indeed, unconditionally 1-private computation of arbitrary functions is possible whenever the number of players is at least 3 [6,9].

The above question is motivated by several open problems in the area of MPC. One such problem is related to the possibility of secure computation in the following natural scenario. Suppose that n clients wish to employ m servers in order to securely compute some (possibly complex) function of their inputs. We would like to obtain a non-interactive protocol in which each client sends a single message to each server, depending on its input and its local randomness, and gets a single message from each server in return. The protocol should protect uncorrupted clients from any collusion involving corrupted clients and a single server. (This can be generalized to collusions

[3] Allowing a small gap between the privacy threshold and the reconstruction threshold, the algebraic–geometric secret sharing scheme of Chen and Cramer [10] can support d-multiplication with the additional feature that the size of each share does not grow with the number of players.

involving clients and at most t servers.) Current techniques allow solving this problem with $m \geq 4$ servers and leave the possibility of using only 3 servers open. The existence of a 1-private 3-multiplicative secret sharing scheme for 3 players would imply that 3 servers are indeed sufficient.

Other open problems which motivate our main question include the exact characterization of the round complexity of secure multiparty computation of general functions and the communication complexity of securely computing low-degree multivariate polynomials, constant-depth circuits, and natural database search functionalities (see Sect. 3 for a discussion of these problems). A positive answer to the main question, even one obtained via *nonlinear* secret sharing, would imply solutions to these open problems.

Our main result provides a negative answer to the question, implying a limitation on the usefulness of secret sharing in the context of MPC. More generally, there exists a d-multiplicative secret sharing scheme realizing a given access structure if *and only if* the access structure is of type Q_d. It is interesting to note that in the case $d = 3$ this is precisely the condition for strong multiplication.

1.1. *Technique*

Interestingly, the proof of our main result relies on a *quantitative* argument inspired by lower bound techniques in the area of communication complexity. To the best of our knowledge, this is the first time such techniques are used in the context of proving negative results on the feasibility of secret sharing or secure computation. (Different connections between communication complexity and secret sharing were recently exploited in [4].) More concretely, we show that it is impossible to implement a secret sharing scheme such that (i) no individual player gets information about the secret from its share, and (ii) all d players can locally convert shares of d different secrets into an additive representation of the product of these secrets. This impossibility result implies the characterization described above. The proof shows a method for d servers, holding a vector y of n field elements, to use any d-multiplicative secret sharing scheme in order to communicate y to a client by sending him less than n field elements altogether. This method employs a technique that was previously used by Babai et al. [2] in the context of obtaining lower bounds for the simultaneous messages model in communication complexity.

The high level idea is the following. Let $m = O(n^{1/d})$ and let $V = \{v_1, \ldots, v_n\}$ be a set of n distinct length-m vectors, each containing the value 1 in d positions and the value 0 elsewhere. The vector y can be represented by an m-variate degree-d polynomial p_y such that $p_y(v_j) = y_j$ for each $1 \leq j \leq n$. By the d-multiplication property of the given secret sharing scheme, if the vector v_j is (componentwise) secret-shared between the d servers, the servers can non-interactively communicate $y_j = p_y(v_j)$ to the client by each sending a single field element which depends on its share of v_j. The crucial property of our particular choice of V is that it is possible to find a valid secret-sharing of all n vector v_j such that the total number of *distinct* shares given to each server is only $O(n^{(d-1)/d}) = o(n)$ rather than n. Thus, the entire vector y can be communicated to the client by having each server send only $o(n)$ field elements, which for sufficiently large n yields the desired contradiction.

Related Work In a recent and independent work, Zhang et al. [30] studied the power of d-multiplicative *linear* secret sharing schemes and their relation with strongly multiplicative schemes. In particular, they showed that any linear 3-multiplicative scheme is also strongly multiplicative, and that any linear strongly multiplicative scheme can be efficiently converted into a strongly multiplicative scheme for the same access structure. As discussed above, this answers our main question for the case of linear schemes, but leaves open the question for the general case.

Organization In Sect. 2, we define d-multiplicative secret sharing schemes and discuss some of their properties. In Sect. 3, we present open problems and applications which motivate the main question we address in this work. Finally, in Sect. 4, we present our main (negative) result, which together with a folklore positive result (proved in Appendix A) gives a precise characterization of the access structures for which d-multiplicative secret sharing is possible.

2. Preliminaries

In this section, we formally define the notion of d-multiplicative secret sharing and discuss some of its basic properties.

A secret sharing scheme involves a dealer and k players P_i, $1 \leq i \leq k$. We consider secret sharing schemes in which the secret is taken from a finite field $\mathbb{F}$.[4] The scheme specifies a randomized mapping from the secret s to a k-tuple of shares $\langle s_1, \ldots, s_k \rangle$, where the share s_i is given to player P_i. We assume that all shares s_i are taken from a finite share domain $\mathcal{S}$, and let $\mathcal{D}$ denote a discrete probability distribution from which the dealer's randomness is chosen. To share a secret $s \in \mathbb{F}$, the dealer chooses a random string $r \in \mathcal{D}$ and applies a sharing function $\textsc{Share} : \mathbb{F} \times \mathcal{D} \to \mathcal{S}^k$ to compute $\textsc{Share}(s, r) = \langle s_1, \ldots, s_k \rangle$. For $T \subset [k]$, we let $\textsc{Share}(s, r)_T$ denote the restriction of $\textsc{Share}(s, r)_T$ to its T-entries.

We turn to define the t-privacy and d-multiplication properties of secret sharing schemes. We say that a secret sharing scheme is t-private if no set of t players can learn anything about the secret from their shares. In contrast to traditional secret sharing, here we do not insist that every set of $t + 1$ players should be able to completely reconstruct the secret from their shares. (This makes our negative results stronger.) However, the multiplication property implies that all players together can reconstruct the secret.

Definition 2.1 (t-Private secret sharing). A secret sharing scheme is said to be *t-private* if for every pair of secrets $s, s' \in \mathbb{F}$ and every set $T \subseteq [k]$ such that $|T| = t$, the random variables $\textsc{Share}(s, r)_T$ and $\textsc{Share}(s', r)_T$ (induced by a random choice of $r \in \mathcal{D}$) are identically distributed.

We now define the d-multiplication property of secret sharing schemes which is the focus of this work. In a d-multiplicative secret sharing scheme, each player should be able to apply a local computation on its shares of d secrets, such that the outcomes of the k local computations always add up to the product of the d secrets. Formally:

[4] Our definitions and results can be generalized to the case of d-multiplicative secret sharing over arbitrary finite *rings*.

Definition 2.2 (d-Multiplicative secret sharing). We call a secret sharing scheme *d-multiplicative* if it satisfies the following d-multiplication property. Let $s^1, \ldots, s^d \in \mathbb{F}$ be d secrets, and $r^1, \ldots, r^d$ be d elements in the support of $\mathcal{D}$. For $1 \le j \le d$, let $\langle s_1^j, \ldots, s_k^j \rangle = \textsc{Share}(s^j, r^j)$. We require the existence of a function $\textsc{Mult} : [k] \times \mathcal{S}^d \to \mathbb{F}$ such that for all possible s^j and r^j as above we have $\sum_{i=1}^k \textsc{Mult}(i, s_i^1, \ldots, s_i^d) = \prod_{j=1}^d s^j$.

Remark 2.3. Note that the multiplication property imposes no linearity requirement on the secret sharing scheme itself. That is, it may be that given shares of two secret $s_1, s_2 \in \mathbb{F}$, the players have no way of computing valid shares of the secret $s_1 + s_2$ (with respect to the same scheme). This should not be confused with the ability to generate an additive sharing of products of secrets after applying the function $\textsc{Mult}$.

In applications of d-multiplicative secret sharing that will be presented in the next section, we will use the multiplication property to evaluate multivariate polynomials on vectors of shared secrets. This motivates the following definition.

Definition 2.4 (Evaluating a polynomial on shares). Let $p \in \mathbb{F}[x_1, \ldots, x_m]$ be an m-variate polynomial over $\mathbb{F}$ that can be written as the sum of degree-d monomials (of the form $\alpha \cdot x_{i_1} x_{i_2} \cdots x_{i_d}$). We define the operation of p on a vector of secret shares $\langle s_i^1, \ldots, s_i^m \rangle \in \mathcal{S}^m$, where the shares are held by player P_i, for some $i \in [k]$. Each monomial is evaluated by applying the function $\textsc{Mult}$ to the corresponding shares, and addition is simply the addition over $\mathbb{F}$. That is, if

$$p(x_1, \ldots, x_m) = \sum_{\substack{J \in [m]^d \\ J = (j_1, \ldots, j_d)}} \alpha_J \prod_{\ell=1}^{d} x_{j_\ell}$$

then

$$p_i\left(s_i^1, \ldots, s_i^m\right) \stackrel{\text{def}}{=} \sum_{\substack{J \in [m]^d \\ J = (j_1, \ldots, j_d)}} \alpha_J \cdot \textsc{Mult}\left(i, s_i^{j_1}, \ldots, s_i^{j_d}\right).$$

The above definition can be naturally extended to general polynomials of total degree (at most) d by converting each monomial of degree $d' < d$ into an equivalent monomial of degree d. This conversion is done by padding the monomial with $d - d'$ copies of a dummy variable x_0, whose corresponding secret will be set to 1. The shares of this secret will always be set to $\textsc{Share}(1, r_0)$, where r_0 is some fixed element in the support of $\mathcal{D}$.

The following straightforward lemma shows that d-multiplication can be used to locally convert shares of the inputs of a degree-d multivariate polynomial into additive shares of its output.

Lemma 2.5. *Let $p \in \mathbb{F}[x_1, \ldots, x_m]$ be a degree d polynomial over $\mathbb{F}$. Suppose that the vector of secrets $\langle s^1, \ldots, s^m \rangle \in \mathbb{F}^m$ was coordinate-wise secret shared using a d-*

multiplicative secret sharing scheme, such that for every $j \in [m]$, *the shares corresponding to* s^j *are* $\langle s_1^j, \ldots, s_k^j \rangle \in \mathcal{S}^k$. *Then, it holds that*

$$p\left(s^1, \ldots, s^m\right) = \sum_{i=1}^{k} p_i\left(s_i^1, \ldots, s_i^m\right).$$

3. Motivating Applications

In this section, we highlight several open problems in the area of MPC and connect them with the problem of d-multiplicative secret sharing. These connections motivate our main result, which rules out a solution of the open problems by means of better d-multiplicative secret sharing schemes.

Since this section is mainly intended for motivational purposes, we do not include formal definitions of the (standard) notions of MPC we will be using, and refer readers to the literature (e.g., [8], [14, Chap. 7]) for such definitions.

3.1. *Secure Polynomial Evaluation*

In all of the following applications, we will apply a d-multiplicative secret sharing scheme in order to securely evaluate a multivariate polynomial of total degree (at most) d. For simplicity, we restrict the attention to security against "honest but curious" players, who follow the protocol's instructions but try to learn as much as possible about other players' inputs from the messages they receive. This is also referred to as security in the *semi-honest* model. We note, however, that the following protocols and questions can be extended to the case of security against malicious players who may behave arbitrarily.

In presenting the following secure polynomial evaluation protocol, it will be convenient to distinguish between "clients", who hold inputs and receive outputs, and "servers" who help perform the computation and only know the identity of the polynomial that should be evaluated. Since the two roles are not mutually exclusive, this generalizes the standard setting for secure computation in which each player is both a client and a server.

More concretely, we consider a scenario in which n clients jointly hold inputs $\langle s^1, \ldots, s^m \rangle \in \mathbb{F}^m$ (where each input s^i is known to only one of the clients) and wish to evaluate a publicly known degree-d polynomial p on their joint inputs without revealing their inputs to each other. To this end, the clients can interact with k servers. The protocol should satisfy the following correctness and privacy requirements:

- *Correctness.* All clients output $p(s^1, \ldots, s^m)$ (assuming that both clients and servers follow the protocol).
- t*-Privacy.* Any collusion involving a strict subset of the clients and at most t servers should not learn anything about the inputs of the other clients other than what follows from their own inputs and the output.

For simplicity, we assume that m is a multiple of n and Client i holds inputs $(s^{(i-1)\ell+1}, \ldots, s^{i\ell})$, where $\ell \stackrel{\text{def}}{=} m/n$.

The following simple protocol applies d-multiplicative secret sharing for solving the above secure polynomial evaluation problem with only two rounds of interaction.

Lemma 3.1. *Given a t-private d-multiplicative secret sharing for k players over $\mathbb{F}$, there exists a t-private k-server secure polynomial evaluation protocol as above for m-variate polynomials of degree d over $\mathbb{F}$. The protocol requires only two rounds of interaction, and its communication complexity[5] is $O(m)$.*

The protocol proceeds as follows:

- *Round 1:* Client j, $1 \leq j \leq n$, shares every input s^h he holds by computing $\text{SHARE}(s^h, r^h) = \langle s_1^h, \ldots, s_k^h \rangle$. After sharing his ℓ inputs, he sends the vector of shares that correspond to Server i, $\langle s_i^{(j-1)\ell+1}, \ldots, s_i^{j\ell} \rangle$ to Server i. In addition, Client j distributes between the servers random additive shares of 0, namely it sends to Server i a field element z_i^j such that the k elements z_i^j are random subject to the restriction that they add up to 0.
- *Round 2:* Server i, $1 \leq i \leq k$, computes $y_i = p_i(s_i^1, \ldots, s_i^m) + \sum_{j=1}^{n} z_i^j$ (where $p_i(s_i^1, \ldots, s_i^m)$ is as defined in Definition 2.4), and sends y_i to all clients.
- *Outputs:* Each client computes and outputs $\sum_{i=1}^{k} y_i$. By Lemma 2.5, this output is equal to $p(s^1, \ldots, s^m)$ as required.

In the following sections, we apply the above protocol in several different MPC scenarios.

3.2. *Secure Computation with Minimal Interaction*

The above secure polynomial evaluation protocol has the appealing feature of requiring a minimal amount of interaction: the protocol involves only a single message from each client to each server, followed by a single message from each server to each client. The following open questions address the feasibility of *general* secure computation with such a minimal amount of interaction.

Question 1. What is the smallest number of servers k such that n clients can t-privately evaluate an *arbitrary* function f of their inputs by sending a single message to each server and receiving a single message in return? In particular, do 3 servers suffice for 1-privacy?

The best upper bound on the number of servers k implied by current techniques (see below) is $k = 3t + 1$. In particular, the best known 1-private protocol requires 4 servers (regardless of the number of clients). The above question is open even if one settles for computational (as opposed to unconditional, or "information-theoretic") privacy.

We now formulate a variant of Question 1 that applies to the standard MPC model, in which k players wish to privately compute some function on their inputs by directly communicating with each other over point-to-point channels.

[5] The size of the field $\mathbb{F}$ and the number of players are considered as constants.

Question 2. What is the maximal privacy level t for which k players can evaluate an arbitrary function f on their inputs using a t-private *2-round* MPC protocol? In particular, does every k-argument function f have a 2-round, $\lfloor \frac{k-1}{2} \rfloor$-private MPC protocol?

It is known [18] that a privacy level of $t = \lfloor \frac{k-1}{3} \rfloor$ can be achieved (see below), or alternatively $t = \lfloor \frac{k-1}{2} \rfloor$ can be achieved if one allows three (rather than two) rounds of interaction. The latter level of privacy is optimal if one insists on unconditional privacy [12]. Thus, while it is possible to get optimal privacy with nearly optimal interaction or optimal interaction with nearly optimal privacy, the possibility of simultaneously maximizing privacy and minimizing interaction remains an intriguing open question.

We now relate these questions to the problem of d-multiplicative secret sharing. This relation together with standard constructions of d-multiplicative secret sharing schemes (e.g., Shamir's scheme for $k > dt$ players) implies the state of the art mentioned above.

Using *randomizing polynomials* [18] it is possible to represent an arbitrary function f by a vector of (randomized) degree-3 polynomials. The complexity of such a representation can be polynomial in the branching program size of f [19], or even in the circuit size of f if one settles for computational privacy [1] (and assumes the existence of a pseudorandom generator in NC^1). Combining randomizing polynomials with Lemma 3.1 we thus have the following:

Claim 3.2. *Suppose there is a t-private 3-multiplicative secret sharing for k players. Then there is a t-private solution to Question 1 (resp., Question 2) with k servers (resp., players). In both cases, the complexity of the protocol is either polynomial in the branching program size of f (with perfect privacy) or in the circuit size of f (with computational privacy, assuming the existence of a pseudorandom generator in NC^1).*

In particular, the existence of a 3-multiplicative secret sharing scheme for k players that is $\lfloor \frac{k-1}{2} \rfloor$-private (as opposed to $\lfloor \frac{k-1}{3} \rfloor$-private, which is currently known) would imply a positive answer to both open questions. Unfortunately, in Sect. 4, we show that the latter is impossible to achieve.

3.3. *Communication Complexity of Secure Computation*

The polynomial evaluation protocol from Sect. 3.1 also suggests that better d-multiplicative schemes would imply progress on the following open question:

Question 3. What is the minimal number of servers/players k required for t-private computation of degree-d multivariate polynomials in which the communication complexity is linear in the length of the input (rather than the number of monomials)?

Currently it is known that $k > dt/2$ servers suffice [3,29]. The existence of a t-private d-multiplicative secret sharing scheme for k players implies (via a straightforward use of Lemma 3.1) that k players suffice.

Progress on Question 3 is motivated by techniques for representing useful classes of functions by *short* vectors of low-degree polynomials, see [3,26,28]. Using such representations, progress on Question 3 would imply better sublinear-communication pro-

tocols for securely evaluating DNF and CNF formulas, securely searching for partial matches, and solving other natural database search problems.

4. The Negative Result

In this section, we give an exact characterization of the type of access structures for which d-multiplicative secret sharing is possible.

Our main result rules out the existence of a 1-private d-multiplicative secret sharing schemes involving d (or less) players, for any value of d. The proof of this result uses an information theoretic argument borrowed from the field of communication complexity. Suppose there are d servers holding a vector $y \in \mathbb{F}^n$. The servers want to communicate y to a client. We show that given a d-multiplicative secret sharing scheme for d players that is 1-private, the servers can do so by sending to the client less than n field elements (for sufficiently large n) and by this we derive a contradiction. For this we use the decomposition technique of Babai et al. [2], which utilizes a small number of information pieces to reconstruct a large amount of data. The high level idea of the proof was already described in Sect. 1.1. Below we provide the details.

Lemma 4.1. *There is no secret sharing scheme for d players that is 1-private and d-multiplicative.*

Proof. For notational convenience, we present the proof for the case $d = 3$. The generalization to an arbitrary $d > 3$ is straightforward. Suppose there is a 1-private, 3-multiplicative secret sharing scheme for 3 players over some finite field $\mathbb{F}$, with share domain $\mathcal{S}$. Suppose S_1, S_2 and S_3 are three servers holding a vector $y \in \mathbb{F}^n$. The servers are interested in communicating y to a client, using a small amount of communication. Clearly, they cannot do that by sending less than n field elements. We show that given the 3-multiplicative 1-private secret sharing scheme they can do it more efficiently (for a sufficiently large n), and by this we derive a contradiction to the existence of the scheme.

Let $\langle s_1^0, s_2^0, s_3^0 \rangle$ be a valid secret sharing of the secret 0. By the 1-privacy requirement, there must exist two shares s_2^1, and s_3^1, such that $\langle s_1^0, s_2^1, s_3^1 \rangle$ is a valid secret sharing of the secret 1 (otherwise, given that $\text{SHARE}(s, r)_1 = s_1^0$, player P_1 knows that the secret 1 is impossible while the probability of the secret 0 is positive). Similarly, there exist shares s_1^2, s_3^2 such that $\langle s_1^2, s_2^0, s_3^2 \rangle$ is a valid secret sharing of 1, and shares s_1^3, s_2^3 such that $\langle s_1^3, s_2^3, s_3^0 \rangle$ is a valid secret sharing of 1.

Let n be a sufficiently large integer ($n > 1200$ will do) such that $n = \binom{m}{3}$ for some positive integer m. Let $V = \{v_1, \ldots, v_n\}$ be the set of all distinct length-m vectors containing the value 1 in three positions and the value 0 elsewhere. Let $y \in \mathbb{F}^n$ be a vector of length n. The vector y can be represented by an m-variate degree-3 polynomial p such that $p(v_j) = y_j$ for each $1 \leq j \leq n$. Indeed, letting $h_{j,1}, h_{j,2}, h_{j,3}$ denote the coordinates in which v_j is equal to 1, we can define

$$p(x_1, \ldots, x_m) = \sum_{j=1}^{n} y_j \cdot x_{h_{j,1}} x_{h_{j,2}} x_{h_{j,3}}.$$

That is, p is a degree 3 polynomial encoding of y.

Define the set $Q_1 \subseteq \mathcal{S}^m$ as follows. The set Q_1 contains all the vectors $q \in \mathcal{S}^m$ of the following form: There are $m-2$ entries with the value s_1^0, one entry with the value s_1^2, and one entry with the value s_1^3. All together, there are $m(m-1) = O(n^{2/3})$ vectors in Q_1. Similarly, we define the set $Q_2 \subseteq \mathcal{S}^m$ containing the vectors whose entries are all equal to s_2^0 except one entry that equals s_2^1 and one that equals s_2^3, and the set $Q_3 \subseteq \mathcal{S}^m$ containing the vectors with all entries equal to s_3^0 except one that equals s_3^1 and one that equals s_3^2.

We are now ready to define the messages that the servers send to the client. The message z_i sent by server i contains, for each $q \in Q_i$, the value of $p_i(q)$ as defined in Definition 2.4. Note that each message contains exactly $m(m-1)$ field elements.

We claim that given z_1, z_2 and z_3, the client can completely reconstruct y. For $1 \leq j \leq n$, the client computes y_j as follows. By the definitions of V and p, it suffices for the client to compute $p(v_j)$, where v_j is the length-m characteristic vector of the set $\{h_{j,1}, h_{j,2}, h_{j,3}\}$. Consider the vector $q_1 \in Q_1$ in which the $h_{j,2}$ entry is s_1^2, the $h_{j,3}$ entry is s_1^3, and all other entries are s_1^0. Similarly, let $q_2 \in Q_2$ be the vector in which the $h_{j,1}$ entry is s_2^1, the $h_{j,3}$ entry is s_2^3, and all other entries are s_2^0. Finally, let $q_3 \in Q_3$ be the vector in which the $h_{j,1}$ entry is s_3^1, the $h_{j,2}$ entry is s_3^2 and all other entries are s_3^0.

Note that taking the $h_{j,1}$ entry of the three vectors $\langle q_1, q_2, q_3 \rangle$, we get the shares $\langle s_1^0, s_2^1, s_3^1 \rangle$ which is a valid secret sharing of 1. Similarly, in the $h_{j,2}$ entry we get $\langle s_1^2, s_2^0, s_3^2 \rangle$ and in the $h_{j,3}$ entry we get $\langle s_1^3, s_2^3, s_3^0 \rangle$, which are both secret sharings of 1. In every other entry, we get $\langle s_1^0, s_2^0, s_3^0 \rangle$ which is a valid secret sharing of 0. It follows that the vectors q_1, q_2, and q_3 form a valid coordinate-wise secret sharing of v_j, and thus by Lemma 2.5 we have $p_1(q_1) + p_2(q_2) + p_3(q_3) = p(v_j)$.

We conclude that the servers can communicate any $y \in \mathbb{F}^n$ to the client using only $3m(m-1)$ field elements. Whenever $n = \binom{m}{3} > 3m(m-1)$ this is impossible, and thus we get a contradiction to the existence of the desired secret sharing scheme. □

Remark 4.2 (Extensions). The proof of Lemma 4.1 can be extended to the case of d-multiplicative secret sharing in which secrets are taken from an arbitrary finite *ring* (as opposed to a field). It can also be directly extended to the case of *statistical* secret sharing schemes in which both the correctness and privacy requirements are relaxed to allow some small statistical error. We do not know how to extend the negative result to the case of *computationally* private secret sharing schemes and leave this as an open question.

4.1. *A General Characterization*

Lemma 4.1 implies a more general negative result, ruling out the existence of a t-private d-multiplicative secret sharing scheme for dt players. Below we further generalize this result from the threshold case to general access structures. Together with a (folklore) matching positive result, which we prove in Appendix A, this provides a complete characterization of the access structures for which d-multiplication is possible.

While the usual notion of access structure specifies a collection of authorized player sets, here it will be more convenient to use the complementary notion of an *adversary structure*, specifying a collection of *unauthorized* sets.

Definition 4.3 (Adversary structure). A k-player *adversary structure* is a collection of sets $\mathcal{T} \subseteq 2^{[k]}$ that is closed under subsets; that is, if $T \in \mathcal{T}$ and $T' \subseteq T$ then $T' \in \mathcal{T}$.

The definition of t-privacy from Sect. 2 naturally generalizes to $\mathcal{T}$-privacy for a general adversary structure $\mathcal{T}$. Formally:

Definition 4.4 ($\mathcal{T}$-Private secret sharing). Let $\mathcal{T}$ be an adversary structure. A secret sharing scheme is said to be $\mathcal{T}$-*private* if for every pair of secrets $s, s' \in \mathbb{F}$ and every $T \in \mathcal{T}$, the random variables $\text{SHARE}(s, r)_T$ and $\text{SHARE}(s', r)_T$ (induced by a random choice of $r \in \mathcal{D}$) are identically distributed.

We now define a property of adversary structures that will be useful for our main characterization.

Definition 4.5 (Adversary structure of type Q_d). Let d, k be positive integers and $\mathcal{T}$ be a k-player adversary structure. We say that $\mathcal{T}$ is of *type* Q_d if for every d sets $T_1, \ldots, T_d \in \mathcal{T}$ we have $T_1 \cup T_2 \cup \cdots \cup T_d \subset [k]$; that is, no d unauthorized sets cover the entire set of players.

Theorem 4.6. *For any positive integers k, d and a k-player adversary structure $\mathcal{T}$, there exists a d-multiplicative $\mathcal{T}$-private secret sharing scheme if and only if $\mathcal{T}$ is of type Q_d.*

Proof. If $\mathcal{T}$ is not of type Q_d, then the set of players $[k]$ can be partitioned into d disjoint subsets $T_1, \ldots, T_d \in \mathcal{T}$. This gives rise to a 1-private d-multiplicative secret sharing scheme on d players, where Player i in the new scheme gets the shares of all players in the set T_i in the original scheme, in contradiction with Lemma 4.1. For the other direction, note that the CNF secret sharing scheme (cf. [5,22,24]) is d-multiplicative for any adversary structure of type Q_d. See Appendix A for details. □

Appendix A. A Positive Result

In this section, we prove a (folklore) positive result on d-multiplicative secret sharing which matches our negative result. Specifically, we show that every adversary structure of type Q_d can be realized by a d-multiplicative scheme. (Recall that an adversary structure is of type Q_d if no d unauthorized sets cover the entire set of players.) This result will be proved by using the following specific secret sharing scheme.

Definition A.1 (CNF secret sharing) [22]. Let $\mathcal{T}$ be a k-player adversary structure. The $\mathcal{T}$-*private CNF secret sharing scheme* is defined by the following sharing algorithm. Let $\hat{\mathcal{T}}$ be the collection of maximal sets in $\mathcal{T}$ (namely those that are not contained in any other set from $\mathcal{T}$). The dealer first additively breaks s into $|\hat{\mathcal{T}}|$ additive parts r_T, $T \in \hat{\mathcal{T}}$. (That is, the parts r_T are chosen at random from $\mathbb{F}$ subject to the restriction that their sum is s.) The share of player P_i consists of all parts r_T such that $i \notin T$.

Note that in the t-private CNF scheme each player P_i receives exactly $\binom{k-1}{t}$ field elements, namely the parts r_T labeled by the sets $T \in \binom{[k]}{t}$ which do not contain i. The t-privacy property follows by observing that every set T of k players jointly misses the share r_T, and thus can learn no information about T.

We are now ready to state the positive result, which extends in a simple way previous applications of the CNF scheme in the context of MPC [5,24].

Theorem A.2. *For any adversary structure $\mathcal{T}$ of type Q_d, the $\mathcal{T}$-private CNF secret sharing scheme is d-multiplicative.*

Proof. For $1 \leq j \leq d$, we let r_T^j, $T \in \hat{\mathcal{T}}$, denote the additive parts of secret s^j. Writing the product $s^1 \cdots s^d$ as the sum of the $|\hat{\mathcal{T}}|^d$ monomials of the form $r_{T_1}^1 \cdots r_{T_d}^d$, we can partition the monomials into k sets X_i such that all monomials in set X_i are known to P_i. This follows from the fact that every monomial as above can be assigned to a set X_i such that $i \notin T_1 \cup \cdots \cup T_d$ (the existence of such i follows from the assumption that $\mathcal{T}$ is of type Q_d). The d-multiplication property follows by letting $\textsc{Mult}(i, \cdot)$ output the sum of all monomials in X_i. □

Note that the CNF secret sharing scheme is generally inefficient, and this is also the case in the important case of threshold structures. However, for this case the t-private secret sharing scheme of Shamir [27] can be used as an efficient d-multiplicative secret sharing scheme whenever $k > dt$.

References

[1] B. Applebaum, Y. Ishai, E. Kushilevitz, Computationally private randomizing polynomials and their applications. *Comput. Complex.* **15**(2), 115–162 (2006). Earlier version in *Proc. CCC '05*

[2] L. Babai, A. Gál, P.G. Kimmel, S.V. Lokam, Communication complexity of simultaneous messages. *SIAM J. Comput.* **33**(1), 137–166 (2003). Earlier version in *Proc. STACS '95*

[3] O. Barkol, Y. Ishai, Secure computation of constant-depth circuits with applications to database search problems, in *Proc. CRYPTO '05* (2005), pp. 395–411

[4] O. Barkol, Y. Ishai, E. Weinreb, Communication in the presence of replication, in *Proc. 40th STOC* (2008), pp. 661–670

[5] D. Beaver, A. Wool, Quorum-based secure multi-party computation, in *Proc. EUROCRYPT '98* (1998), pp. 375–390

[6] M. Ben-Or, S. Goldwasser, A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, in *Proc. 20th STOC* (1988), pp. 1–10

[7] G.R. Blakley, Safeguarding cryptographic keys, in *Proc. of the American Federation of Information Processing Societies (AFIPS)*, vol. 48 (1979), pp. 313–317

[8] R. Canetti, Security and composition of multiparty cryptographic protocols. *J. Cryptol.* **13**(1), 143–202 (2000)

[9] D. Chaum, C. Crépeau, I. Damgård, Multiparty unconditionally secure protocols, in *Proc. 20th STOC* (1988), pp. 11–19

[10] H. Chen, R. Cramer, Algebraic geometric secret sharing schemes and secure multi-party computations over small fields, in *CRYPTO* (2006), pp. 521–536

[11] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, V. Vaikuntanathan, Secure computation from random error correcting codes, in *EUROCRYPT* (2007), pp. 291–310

[12] B. Chor, E. Kushilevitz, A zero-one law for boolean privacy. *SIAM J. Discrete Math.* **4**(1), 36–47 (1991)

[13] R. Cramer, I. Damgård, U. Maurer, General secure multi-party computation from any linear secret-sharing scheme, in *Proc. EUROCRYPT '00* (2000), pp. 316–335
[14] O. Goldreich, *Foundations of Cryptography: Vol. 2, Basic Applications* (Cambridge University Press, New York, 2004)
[15] O. Goldreich, S. Micali, A. Wigderson, How to play any mental game, in *Proc. 19th STOC* (1987), pp. 218–229
[16] D. Harnik, Y. Ishai, E. Kushilevitz, J.B. Nielsen, Ot-combiners via secure computation, in *TCC* (2008), pp. 393–411
[17] M. Hirt, U. Maurer, Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptol.* **13**(1), 31–60 (2000). Earlier version in *Proc. PODC '97*
[18] Y. Ishai, E. Kushilevitz, Randomizing polynomials: A new representation with applications to round-efficient secure computation, in *Proc. 41st FOCS* (2000), pp. 294–304
[19] Y. Ishai, E. Kushilevitz, Perfect constant-round secure computation via perfect randomizing polynomials, in *Proc. 29th ICALP* (2002), pp. 244–256
[20] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai, Zero-knowledge from secure multiparty computation, in *STOC* (2007), pp. 21–30
[21] Y. Ishai, M. Prabhakaran, A. Sahai, Founding cryptography on oblivious transfer—efficiently, in *CRYPTO* (2008), pp. 572–591
[22] M. Ito, A. Saito, T. Nishizeki, Secret sharing schemes realizing general access structure, in *Proc. of the IEEE Global Telecommunication Conf., Globecom 87* (1987), pp. 99–102. Journal version: Multiple assignment scheme for sharing secret. *J. Cryptol.* **6**(1), 15–20 (1993)
[23] E. Kushilevitz, Privacy and communication complexity. *SIAM J. Discrete Math.* **5**(2), 273–284 (1992)
[24] U.M. Maurer, Secure multi-party computation made simple. *Discrete Appl. Math.* **154**(2), 370–381 (2006). Earlier version in *Proc. SCN '02*
[25] I.C. Pueyo, H. Chen, R. Cramer, C. Xing, Asymptotically good ideal linear secret sharing with strong multiplication over *ny* fixed finite field, in *CRYPTO* (2009), pp. 466–486
[26] A. Razborov, Lower bounds for the size of circuits of bounded depth with basis (AND, XOR). *Math. Notes Acad. Sci. USSR* **41**(4), 333–338 (1987)
[27] A. Shamir, How to share a secret. *Commun. ACM* **22**(11), 612–613 (1979)
[28] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, in *STOC* (1987), pp. 77–82
[29] D.P. Woodruff, S. Yekhanin, A geometric approach to information-theoretic private information retrieval. *SIAM J. Comput.* **37**(4), 1046–1056 (2007). Earlier version in *Proc. CCC '05*
[30] Z. Zhang, M. Liu, Y.M. Chee, S. Ling, H. Wang, Strongly multiplicative and 3-multiplicative linear secret sharing schemes, in *ASIACRYPT* (2008), pp. 19–36