COMPUTING FUNCTIONS OF A SHARED SECRET

AMOS BEIMEL*, MIKE BURMESTER†, YVO DESMEDT‡, AND EYAL KUSHILEVITZ§

Abstract. In this work we introduce and study threshold (t-out-of-n) secret sharing schemes *for families of functions* $\mathcal{F}$. Such schemes allow any set of at least t parties to compute privately the value $f(s)$ of a (previously distributed) secret s, for any $f \in \mathcal{F}$. Smaller sets of players get no more information about the secret than what follows from the value $f(s)$. The goal is to make the shares as short as possible. Results are obtained for two different settings: we study the case when the evaluation is done on a *broadcast channel without interaction*; and we examine what can be gained by allowing evaluations to be done *interactively via private channels.*

Keywords: Secret Sharing, Broadcast Channel, Private Channels, Interaction, Private Computations.

1. Introduction. Suppose, for example, that we are interested in sharing a secret file among n parties in a way that will later allow any t of the parties to test whether a particular string (not known in the sharing stage) appears in this file. This test should be done without revealing the content of the whole file, or giving any other information about the file. This problem can be viewed as an extension of the traditional definition of threshold (t-out-of-n) secret sharing schemes. In the traditional definition, sets of at least t parties are allowed to reconstruct a (previously distributed) secret s, while any smaller set gets no information about the secret. We introduce a more general definition of *t-out-of-n secret sharing schemes for a family of functions* $\mathcal{F}$. These schemes allow authorized sets of parties to compute some information about the secret and not necessarily the secret itself. More precisely, sets B of size at least t can evaluate $f(s)$ for any function $f \in \mathcal{F}$; any set C of size less than t knows nothing (in the information theoretic sense) about the secret prior to the evaluation of the function; and, in addition, *after* $f(s)$ is computed by a set B no set C of size less than t knows anything more about the secret s than what follows from $f(s)$. In other words, the parties in C might know the value $f(s)$, but they know nothing more than that, although they might have heard part of the communication during the evaluation of $f(s)$.

Clearly, if we consider a family $\mathcal{F}$ that includes only the identity function $f(s) = s$, then we get the traditional notion of secret sharing schemes. These schemes, which were introduced by Blakley [8] and Shamir [34], were the subject of a considerable amount of work (e.g., [30, 26, 28, 6, 35, 20]). They were used in many applications (e.g., [31, 5, 15, 19]) and were generalized in various ways [22, 7, 36]. Surveys are given in [35, 37]. The question of sharing many secrets *simultaneously* was considered (with some differences in the definitions) by several researchers [30, 26, 21, 11, 23, 10, 24]. Simultaneous sharing of many secrets is also a special case of our setting.[1] Other

* Dept. of Mathematics and Computer Science, Ben-Gurion University, Beer Sheva 84105, Israel. E-mail: beimel@cs.bgu.ac.il. http://www.cs.bgu.ac.il/~beimel . This work was done when the author was a Ph.D. student in the Dept. of Computer Science, Technion.

† Dept. of Mathematics, Royal Holloway – University of London, Egham, Surrey TW20 OEX, U.K. E-mail: m.burmester@rhbnc.ac.uk.

‡ Dept. of Computer Science, PO Box 4530, 206 Love Building, Florida State University, Tallahassee, FL 32306-4530, USA. E-mail: desmedt@cs.fsu.edu.

§ Dept. of Computer Science, Technion, Haifa 32000, Israel. E-mail: eyalk@cs.technion.ac.il. http://www.cs.technion.ac.il/~eyalk .

[1] Let $s_1, s_2, \ldots, s_\ell$ be the secrets we want to share *simultaneously*. Construct the concatenated secret $s = s_1 \circ s_2 \circ \ldots \circ s_\ell$ and the functions which can be evaluated are the functions $f_i(s) = s_i$.

similar scenarios in which sharing is viewed as a form of encryption and the security is computationally bounded have been considered in [32, 3, 1].

Threshold cryptography [19, 18, 17] is also a special case of secret sharing for a family of functions.[2] A typical scenario of threshold cryptography is the following: Every set B of t parties should be able to sign any document such that any coalition C of less than t parties cannot sign any other document (even if the coalition C knows signatures of some documents). To achieve this goal the key is shared in such a way that every t parties can generate a signature from their shares without revealing any information on the key except the signature. Specifically, assume we have a signature function $\mathrm{SIGN} : M \times K \to O$, where M is the domain of messages, K is the domain of keys, and O is the domain of signatures. For every $m \in M$ define $f_m : K \to O$ as $f_m(k) = \mathrm{SIGN}(m, k)$. The previous scenario is simply sharing the key for the family $\{f_m : m \in M\}$. These examples show that secret sharing for a family of functions is a natural primitive.

Obviously one possible solution to the problem of sharing a secret for a family $\mathcal{F}$ is by sharing *separately* each of the values $f(s)$ (for any $f \in \mathcal{F}$) using known threshold schemes. While this solution is valid, it is very inefficient, in particular when the size of $\mathcal{F}$ is large. Therefore, an important goal is to realize such schemes while using "small" shares. For example, to share a single bit among n parties, the average (over the parties) length of shares is at least $\log(n - t + 2)$ (for $2 \le t \le n - 1$) [27] and $\log n$ bits are sufficient [34]. The obvious solution for sharing ℓ bits simultaneously will require $\ell \log n$ bits. By [26], it can be shown that shares of at least ℓ bits are necessary. We shall show that ℓ-bit shares are also sufficient if *interactive* evaluation on private channels is allowed, and $O(\ell)$-bit shares are sufficient if non-interactive evaluation (on a broadcast channel) is used.[3]

We present an interactive scheme in which ℓ-bit secrets are distributed using ℓ-bit shares; this scheme allows the computation of every linear combination of the bits (and not only computation of the bits themselves). We use this scheme to construct schemes for other families of functions. The length of the shares in these schemes can be much longer than the length of the secret. An interesting family of functions that we shall consider is the family $\mathcal{ALL}$ of *all* functions of the secret. For this family, we construct a scheme in which the length of the shares is $\approx 2^\ell \log n$ (where ℓ is the length of the secret and 2^ℓ is the length of the description of a function which is evaluated). In this scheme the computation requires no interaction and can be held on a broadcast channel. If we allow interaction on private channels during the computation, then the length of the shares can be reduced to 2^ℓ. Note that the obvious solution of sharing each bit separately requires in this case 2^{2^ℓ} bits.

Our work deals mainly with two models: the *private channels model* in which the computation might require a few rounds of communication; and the *broadcast channel model* in which the computation is non-interactive. On one hand, the broadcast model does not require secure private channels and synchronization; hence, the computation is more efficient. On the other hand, in the private channels model, a coalition C that does not intersect the evaluating set B will know nothing about s; not even the value $f(s)$. Interaction seems to be useful in the computation. It enables us to reduce the

---

The secrets s_i may be dependent: our model allows some information to leak provided it is no more than what follows from the evaluations of $f(s)$.

[2]The functions considered in [19, 18, 17] are, however, very limited and the scenario in [17] is restricted to computational security.

[3]In fact, both results require ℓ to be "sufficiently large": $\ell \ge \log n$ in the interactive case and $\ell \ge \log n \log\log n$ in the broadcast model.

length of the shares by a factor of $\log n$ in our basic scheme for linear functions.

We also study *ideal* threshold schemes. These are schemes in which the size of the shares equals the size of the secrets[4] (see, e.g., [12, 13, 4, 33, 25]). We deal with the characterization of the families of functions $\mathcal{F}$ which can be evaluated by an ideal threshold scheme. For the interactive private channel model, we prove that every boolean function that can be evaluated is a *linear* function. For the broadcast model, we prove that $\mathcal{F}$ cannot contain any boolean function (for every family that contains the identity function).

An example. To motivate our approach and to clarify the notion of secret sharing for a family of functions we consider a simple example, which we discuss informally. Suppose that a dealer shares a secret s among n parties and that at some time later a set of at least t parties would like to answer the question: "Is the shared secret equal to the string a?" That is, they want to evaluate the boolean function $f_a(s)$ which is 1 if and only if $s = a$. The string a is not known when the dealer shares the secret. We describe a simple scheme which makes it possible for the parties to answer such questions. Let $s = s_1 s_2 \dots s_\ell$ be the secret bit-string. The dealer chooses 2ℓ random strings from $\{0,1\}^{2\ell}$, denoted by $b_0, b_1, \dots, b_{2\ell-1}$, and shares each of these strings using Shamir's t-out-of-n threshold scheme [34]. Let $b_{i,j}$ be the share of the secret b_i given to party P_j. The dealer computes the sum (i.e., bitwise exclusive-or) $\alpha = \sum_{i=1}^{\ell} b_{2i+s_i}$ (that is s_i, the value of the i-th bit of the secret, selects whether we take b_{2i} or b_{2i+1}) and gives α to each party. When a subset B of (at least) t parties wants to check if $s = a$ for some $a = a_1 a_2 \dots a_\ell$, each $P_j \in B$ computes a "new" share $\sum_{i=1}^{\ell} b_{(2i+a_i),j}$. Then, the parties in B apply Shamir's secret reconstruction procedure to compute a "new" secret from these "new" shares. Since this procedure involves only computing a linear combination of the shares, the "new" secret is simply $\sum_{i=1}^{\ell} b_{2i+a_i}$. If the "new" secret differs from $\sum_{i=1}^{\ell} b_{2i+s_i}$ then the parties learn that $s \neq a$. However, they get no more information on the secret s. If the "new" secret is equal to $\sum_{i=1}^{\ell} b_{2i+s_i}$ then it is easy to see that with high probability (over the choice of $b_1, \dots, b_\ell$) we must have $s = a$. In this case, the parties learn the secret and there is no additional information which should be hidden from them. Observe that in both cases the parties learn no more than what is revealed by answering the question.

Connection to private computations. In our schemes we require that authorized sets of parties can compute a function of the secret without leaking any other information on the secret. This resembles the requirement of (n,k)-private protocols [5, 15] in which the set of all n parties can evaluate a function of their inputs in a way that no set of less than k parties will gain any additional information about the inputs of the other parties. Indeed, in some of our schemes a set B of size t uses a (t,t)-private protocol. However, it is not necessary to use private protocols in the computation, since the parties are allowed to leak information about their inputs (the shares) as long as they do not leak any additional information on the secret. Moreover, using private protocols does not solve the problem of efficiently sharing a secret for all functions, since not all functions can be computed (t,t) privately [5, 16]. Furthermore, the parties cannot use the $(t, \lfloor (t-1)/2 \rfloor)$-private protocols of [5] or [15] since this means that coalitions of size greater than $t/2$ (but still smaller than t) gain information.

Organization. The rest of this paper is organized as follows. In Section 2 we discuss our model for secret sharing for families of functions and provide the definitions; definitions which are more delicate than in the case of traditional threshold schemes.

[4]The size of each share is always at least the size of the secret [26].

In Section 3 we present interactive and broadcast secret sharing schemes for the family $\mathcal{LIN}$ of linear functions. In Section 4 we use these schemes to construct schemes for other families. In Section 5 we describe the broadcast scheme for the bit functions $\mathcal{BIT}$. In Section 6 we characterize ideal schemes in the various models. Finally, in Section 7 we discuss possible extensions of our work. For completeness we give some background results in Appendices.

2. Definitions. This section contains a formal definition of secret sharing for a family of functions. We start by defining the model. We consider a system with n parties $U = \{P_1, P_2, \ldots, P_n\}$. In addition to the parties, there is a dealer who has a secret input s. A *distribution scheme* is a probabilistic mapping, which the dealer applies to the secret to generate n pieces of information $s_1, \ldots, s_n$ which are referred to as *the shares*. The dealer gives the share s_i to party P_i. Formally,

DEFINITION 2.1 (Distribution Schemes). *Let S be a set of secrets, $S_1, S_2, \ldots, S_n$ sets of shares, and R a set of random inputs. Let $\mu : R \rightarrow [0,1]$ be a probability distribution on R. A* distribution scheme Π *over S is a mapping $\Pi : S \times R \rightarrow S_1 \times S_2 \times \cdots \times S_n$. The i-th coordinate s_i of $\Pi(s,r)$ is the share of P_i. The pair (Π, μ) can be thought of as a dealer who shares a secret $s \in S$ by choosing at random $r \in R$ (according to the distribution μ) and then gives each party P_i as a share the corresponding component of $\Pi(s,r)$.*

In the scenario we consider, the dealer is only active during the initialization of the system. After this stage the parties can communicate. We next define our communication models.

DEFINITION 2.2 (Communication Models). *We consider two models of communication:*

1. *The* private channels *model in which the parties communicate via a complete synchronous network of secure and reliable point-to-point communication channels. In this model a set of parties has access only to the messages sent to the parties in the set.*
2. *The* broadcast channel *model in which the parties communicate via a (public) broadcast channel. In this model a set of parties can obtain all the messages exchanged by the communicating parties.*

A subset of the parties may communicate in order to compute a function of their shares. We now define how they evaluate this function.

DEFINITION 2.3 (Function Evaluation). A set of parties $B \subseteq U$ executes a protocol $F_{B,f}$ to evaluate a function f.

At the beginning of the execution, each party P_i in B has an input s_i and chooses a random input r_i. To evaluate f, the parties in B exchange messages as prescribed by the protocol $F_{B,f}$. In each round every party in B sends messages to every other party in B. A message sent by P_i is determined by s_i, r_i, the messages received by P_i so far, and the identity of the receiver. We say that a protocol $F_{B,f}$ evaluates f (or computes $f(s)$) *if each party in B can always evaluate $f(s)$ from its input and the communication it obtained.*

We denote by $M_C(\langle s_i \rangle_B, \langle r_i \rangle_B)$ the messages that an eavesdropping coalition C can obtain during the communication between the parties in B (this depends on the communication model). Here $\langle s_i \rangle_B$ is the vector of shares of B and $\langle r_i \rangle_B$ the vector of random inputs of B.

A protocol is non-interactive *if the messages sent by each party P_i depend only on its input (and not on the messages received during the execution of the protocol). Non-interactive protocols have only one-round of communication. An* interactive *protocol* *might have more than one round. In this case we require that the protocol terminates after a finite number of rounds (that is, we do not allow infinite runs).*

The parties in the system are honest, that is, they send messages according to the protocol.

A coalition C which eavesdrops on the execution of a protocol $F_{B,f}$ by the parties in B *gains no additional information on the secret* s if any information that C may get is *independent* of the old information. In our model we shall allow C to gain some information, but no more than what follows from the evaluation of f. The information that C gains is determined by the view of C.

DEFINITION 2.4 (The View of a Coalition). *Let $C \subset U$ be a coalition. The* view *of C, denoted $VIEW_C$, after the execution of a protocol consists of the information that C gains. In a distribution scheme $VIEW_C = \langle s_i \rangle_C$, the shares of C. In the evaluation of $f(s)$ by a set of parties $B \subseteq U$ the view of C consists of: the inputs $\langle s_i \rangle_C$, the local random inputs $\langle r_i \rangle_{C \cap B}$ (only the parties in $C \cap B$ are involved in the computation), and the messages that C obtains from the communication channel during the evaluation by B, i.e., $M_{B \cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B)$. That is,*

$$VIEW_C = \langle s_i \rangle_C \circ \langle r_i \rangle_{P_i \in B \cap C} \circ M_{B \cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B).$$

We now define t-out-of-n secret sharing schemes *for a family of functions* $\mathcal{F}$. Such schemes allow any set B of at least t parties to evaluate $f(s)$, for any $f \in \mathcal{F}$, where s is a previously distributed secret, while any set C of less than t parties gains no more information about the secret than the information inferred from $f(s)$. We distinguish between three types of schemes depending on the way that the value $f(s)$ is computed by B and the channels used: (1) *interactive private channels* schemes – where the parties in B engage in a protocol via private channels which computes $f(s)$; (2) *non-interactive private channels* schemes – where the parties in B engage in a non-interactive protocol via private channels to compute $f(s)$; and (3) *broadcast non-interactive* schemes – where each party in B broadcasts a single message (which depends only on its share) on a broadcast channel (we do not consider interactive broadcast schemes in this paper).

The computation of $f(s)$ must be *secure.* There are two requirements to consider: (1) the usual requirement for threshold schemes, that is, before the evaluation any set C of size less than t knows nothing about the secret by viewing their shares; (2) after the evaluation of $f(s)$ by a set B, any set C of size less than t gains no information about s that is not implied by $f(s)$. Formally,

DEFINITION 2.5 (Secret Sharing Schemes for a Family). *A t-out-of-n* secret sharing scheme for a family of functions *$\mathcal{F}$ is a distribution scheme (Π, μ) which satisfies the following two conditions:*

Evaluation. *For any set $B \subseteq U$ of size at least t and any function $f \in \mathcal{F}$ the parties in B can evaluate $f(s)$. That is, there is a protocol $F_{B,f}$ which given the shares of B as inputs will always output the correct value of $f(s)$. The scheme is called* non-interactive *if $F_{B,f}$ is non-interactive, and* interactive *otherwise. Depending on the communication model, a broadcast channel or private channels are used.*

Security. *Let X be a random variable on the set of secrets S and $C \subset U$ be any coalition of size less than t.*

Prior to evaluation (after the distribution of shares).

For any two secrets $s, s' \in S$ and any shares $\langle s_i \rangle_C$ (from $\Pi(s,r)$):

$$\Pr[\, \mathit{VIEW}_C = \langle s_i \rangle_C \mid X = s \,] \quad = \quad \Pr[\, \mathit{VIEW}_C = \langle s_i \rangle_C \mid X = s' \,] \,.$$

After evaluation. *For any $f \in \mathcal{F}$, any $B \subseteq U$ of size at least t, any secrets $s, s' \in S$ with $f(s) = f(s')$, any shares $\langle s_i \rangle_C$, any inputs $\langle r_i \rangle_{B \cap C}$, and any messages $M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B)$ from the computation of $f(s)$ by B:*

$$\begin{aligned} &\Pr[\, \mathit{VIEW}_C = \langle s_i \rangle_C \circ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid X = s \,] \\ &= \Pr[\, \mathit{VIEW}_C = \langle s_i \rangle_C \circ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid X = s' \,] \,. \end{aligned}$$

We say that C gains no information that is not implied by the function f (or simply: gains no additional information) *if in the computation of $f(s)$ we have security after evaluation.*

A (traditional) t-out-of-n secret sharing scheme is a secret sharing scheme for the family that includes only the identity function $f(s) = s$. (As explained in Observation 2.1 below, the family also contains all renamings of the identity function.)

REMARK 2.1. We only require that one function $f \in \mathcal{F}$ can be evaluated securely. A more desirable property is that the scheme is reusable: any number of functions can be evaluated securely (possibly by different sets). All our schemes, except for the scheme in Section 5, are reusable.

REMARK 2.2. In Definition 2.5 we required that all coalitions C of size less than t should gain no additional information on the secret. Clearly, it is sufficient to require this only for coalitions C of size $t-1$, since any smaller coalition has less information on the secret, and, hence, will gain no additional information.[5]

REMARK 2.3. The security in Definition 2.5 is based on the requirement that the view of a coalition is the same for secrets which have the same evaluation. Alternatively we could require: $\Pr[X = s \mid \mathrm{VIEW}_C = \mathit{view}] \;=\; \Pr[X = s \mid f(s) = v]$. That is, the probability that the secret is s, given the view of the coalition C, equals the probability of s, given the value $v = f(s)$. In Appendix A we show that the two definitions are equivalent (see [14] for the analogue claim with respect to traditional secret sharing schemes).

Obviously any broadcast scheme can be transformed into a scheme in which each message is sent on private channels. On the other hand, if we take a non-interactive private channels scheme and broadcast the messages then the security of the computation may be violated.

A function f' is a *renaming* of f if $f(x) = f(y)$ if and only if $f'(x) = f'(y)$. Suppose that f' is a renaming of f and that f can be evaluated securely, then f' can also be evaluated securely. Indeed, to evaluate $f'(s)$ securely we first evaluate $f(s)$ securely; after this evaluation s might not be known. Nevertheless, we can find an input y such that $f(y) = f(s)$, and output $f'(y)$.

OBSERVATION 2.1. *A secret sharing scheme enables the secure evaluation of a* renaming *of f if and only if it enables the secure evaluation of f.*

Therefore, we ignore renamings of functions.

We next define certain classes of functions over $\mathrm{GF}(2^\ell)$ which are of particular interest. Recall the structure of $\mathrm{GF}(2^\ell)$. Each element $a \in \mathrm{GF}(2^\ell)$ is an ℓ-bit string $a_{\ell-1} \ldots a_1 a_0$ which may be represented by the polynomial $a_{\ell-1}x^{\ell-1} + \ldots + a_1 x^1 + a_0$. Addition and multiplication are as for polynomials, using the structure of $\mathrm{GF}(2)$ for

[5] This is different than what happens in private multi-party computation; there a smaller set has less inputs and hence expected to learn even less.

the coefficients and reducing products modulo a polynomial $p(x)$ of degree ℓ, which is irreducible over GF(2). A function $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ is *linear* over GF(2) if $f(x+y) = f(x) + f(y)$ for every $x, y \in \mathrm{GF}(2^\ell)$. Notice, that $f(0) = f(0) + f(0) = 0$ for every linear function f. Thus,

OBSERVATION 2.2. *For every linear function f, for every $x \in \mathrm{GF}(2^\ell)$, and for every $a \in \mathrm{GF}(2)$ it holds that $f(ax) = af(x)$.*

Notice that we *do not* require that $f(ax) = af(x)$ for $a \in \mathrm{GF}(2^\ell)$. We define the family $\mathcal{LIN}_\ell$ to be the family of all linear functions of $\mathrm{GF}(2^\ell)$ over GF(2) (so $\mathcal{LIN}_\ell$ is the family of *additive* functions). Let $e_i : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2)$ be the function that returns the i-th bit of x. Then, $e_i(x+y) = e_i(x) + e_i(y)$ and, hence, e_i is linear. We call the family $\{e_1, \ldots, e_\ell\}$ the *bit functions* and denote it by $\mathcal{BIT}_\ell$. We also consider the family $\mathcal{ALL}_\ell$ of all possible functions of the secret (by Observation 2.1, without loss of generality, the range of the functions in $\mathcal{ALL}_\ell$ is in $\mathrm{GF}(2^\ell)$). We stress that $\mathcal{BIT}_\ell$ contains only boolean functions while both $\mathcal{ALL}_\ell$ and $\mathcal{LIN}_\ell$ contain non-boolean functions as well.

We will need the following claim concerning linear functions.

CLAIM 2.1. *For every linear functions, there are constants $c_1, \ldots, c_\ell$ such that $f(b_1 b_2 \ldots b_\ell) = \sum_{j=1}^{\ell} c_j b_j$ for every ℓ bit string $b_1 b_2 \ldots b_\ell$.*

Proof. Notice that $b_1 b_2 \ldots b_n = \sum_{j=1}^{n} b_j e_j$. Thus, since f is linear, and by Observation 2.2,

$$f(b_1 b_2 \ldots b_n) = f\left(\sum_{j=1}^{n} b_j e_j\right) = \sum_{j=1}^{n} f(b_j e_j) = \sum_{j=1}^{n} f(e_j)\, b_j.$$

Now, define $c_i = f(e_i)$, and the claim follows. □

3. Schemes for the Linear Functions.

3.1. An Interactive Scheme. In this section we show that Shamir's scheme over $\mathrm{GF}(2^\ell)$ [34] (described in Appendix B) is also a secret sharing scheme for the family $\mathcal{LIN}_\ell$ – the family of linear functions.

THEOREM 3.1 (Basic Interactive Scheme). *For every $\ell \geq 1$ there exists an interactive private channels t-out-of-n secret sharing scheme for the family $\mathcal{LIN}_\ell$ in which the secrets have length ℓ and the shares have length $\ell' \triangleq \max\{\ell, \lceil \log(n+1) \rceil\}$.*

Proof. The dealer uses Shamir's t-out-of-n scheme over $\mathrm{GF}(2^{\ell'})$ to distribute the shares. We show how every function $f \in \mathcal{LIN}_\ell$ can be evaluated securely. Consider a set B of $t' \geq t$ parties with shares $\langle s_i \rangle_B$, where $s_i \in \mathrm{GF}(2^{\ell'})$, that wishes to compute $f(s)$. By the reconstruction procedure of Shamir's scheme, for each $P_j \in B$ there exist coefficients δ_j such that $s = \sum_{P_j \in B} \delta_j s_j$ (see Appendix B). Since f is linear, $f(s) = f(\sum_{P_j \in B} \delta_j s_j) = \sum_{P_j \in B} f(\delta_j s_j)$. Let $x_j \triangleq f(\delta_j s_j)$. Then, $f(s)$ is simply the sum of the x_j's. Computing this sum is done using the following interactive protocol of Benaloh [6] (for a more detailed description of this protocol see Appendix C). For convenience of notation, we assume that $B = \{P_1, P_2, \ldots, P_{t'}\}$. In the first step of the protocol each party $P_i \in B$ chooses $t'-1$ random inputs $r_{i,1}, \ldots, r_{i,t'-1}$ in $\mathrm{GF}(2^{\ell'})$, computes $r_{i,t'} \triangleq x_i - \sum_{j=1}^{t'-1} r_{i,j}$, and sends $r_{i,j}$ to $P_j \in B$. Then, each party P_j computes $y_j = \sum_{P_i \in B} r_{i,j}$ and sends y_j to P_1. Party P_1 now computes $\sum_{j=1}^{t'} y_j$ and sends it to all the other parties. By the choice of $r_{i,t'}$, the sum $\sum_{j=1}^{t'} y_j$ equals $\sum_{P_i \in B} x_i = f(s)$, thus, each party in B computes $f(s)$ correctly.

Let us now prove the security of this scheme. The security prior to evaluation follows from the security of Shamir's threshold scheme. For the security after the evaluation it is sufficient to consider a coalition C of size $t-1$ (by Remark 2.2). Since each x_j is computed locally, and since $f(s)$ is computed using a private protocol [6] (for a formal definition of privacy see Appendix C), the parties in C gain no information about the shares of the other parties that is not implied by $f(s)$. That is, for any shares $\langle s_i \rangle_B$ and $\langle s'_i \rangle_B$ with $s_i = s'_i$ for all $P_i \in B \cap C$ and $\sum_{P_j \in B} f(\delta_j s_j) = \sum_{P_j \in B} f(\delta_j s'_j)$, we have:

$$\begin{aligned} &\Pr[\ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid \langle s_i \rangle_B] \\ &\qquad = \Pr[\ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s'_i \rangle_B, \langle r_i \rangle_B) \mid \langle s'_i \rangle_B]. \end{aligned} \tag{1}$$

Now for a secret s and shares $\langle s_i \rangle_C$ with $|C| = t-1$, there are unique shares $\langle s_i \rangle_{B \setminus C}$ of Shamir's scheme that are consistent with s and $\langle s_i \rangle_C$. Therefore,

$$\begin{aligned} \Pr[\ \mathrm{view}_C \mid s\ ] &= \Pr[\ \langle s_i \rangle_C \circ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid s\ ] \\ &= \Pr[\ \langle s_i \rangle_B \circ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid s\ ] \\ &= \Pr[\ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid \langle s_i \rangle_B \circ s\ ] \cdot \Pr[\ \langle s_i \rangle_B \mid s\ ] \\ &= \Pr[\ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid \langle s_i \rangle_B] \cdot \Pr[\ \langle s_i \rangle_B \mid s\ ]\ . \end{aligned} \tag{2}$$

The last equation holds because the shares $\langle s_i \rangle_B$ determine the secret s. By Equation (1), the probability $\Pr[\ \langle r_i \rangle_{B\cap C} \circ M_{B\cap C}(\langle s_i \rangle_B, \langle r_i \rangle_B) \mid \langle s_i \rangle_B]$ is independent of $\langle s_i \rangle_B$ and therefore of s, and by the properties of Shamir's scheme the probability $\Pr[\ \langle s_i \rangle_B \mid s\ ]$ is independent of $\langle s_i \rangle_B$ and s. Therefore, by (2), for every $s' \in S$ such that $f(s) = f(s')$

$$\Pr[\mathrm{view}_C \mid s] = \Pr[\mathrm{view}_C \mid s'],$$

and the coalition C gains no information about s that is not implied by $f(s)$. □

REMARK 3.1. In the special cases when $t = 2$ or $t = n$, the parties can compute $f(s)$ without interaction by simply sending the messages x_j. For $t = 2$ the messages must be sent via private channels. In this case, only coalitions $C \subseteq B$ can get messages of the evaluating set B. Let $C = \{P_i\}$, $B = \{P_i, P_j\}$. Then, party P_i gains no additional information from the message x_j of P_j, because P_i knows $f(s) = x_i + x_j$ and its input x_i, and therefore can compute x_j. For $t = n$ the messages may be sent via a public broadcast channel. In this case, any coalition C is a subset of B, and if C has size $n-1$ then $B = C \cup \{P_j\}$ for some P_j. Again C gains no additional information from x_j, since C can compute x_j from $f(s)$ and $\sum_{P_i \in C} x_i$.

REMARK 3.2. When $3 \le t \le n-1$, the parties evaluate $f(s)$ using an interactive private channels scheme. It can be shown that for this particular scheme interactive evaluation is essential. Otherwise, some coalitions C that intersect the evaluating set B, but are not contained in B, will gain additional information on s when they obtain the values of x_j.

This scheme has the advantage that it is *reusable* – sets $B_1, \ldots, B_m$ can compute values $f_1(s), \ldots, f_m(s)$ while any coalition C does not gain any information on the secret beside the values $f_i(s)$ for every i with $C \cap B_i \neq \emptyset$. This is because the evaluation is done using a private protocol. The private computation assures that C cannot distinguish between two vectors of shares with the same value $f_i(s)$. Hence, even if C knows some evaluations $f_i(s)$, an evaluation of another function will not reveal extra information.

3.2. A Non-interactive Broadcast Scheme. Next, we present a broadcast scheme for the family $\mathcal{LIN}_\ell$ of linear functions, in which the length of the shares is $\ell \log n + \ell$.

THEOREM 3.2 (Basic Broadcast Scheme). *For every $\ell \geq 1$ there exists a non-interactive broadcast t-out-of-n secret sharing scheme for the family $\mathcal{LIN}_\ell$ in which the secrets have length ℓ and the shares have length $\ell \log n + \ell$.*

Proof. Let $q \triangleq \lceil \log(n+1) \rceil$. The dealer shares each bit of the secret independently using Shamir's scheme over $\mathrm{GF}(2^q)$ (this is the smallest possible field of characteristic two for Shamir's scheme). Let $s = b_1 b_2 \ldots b_\ell$ be the secret bit-string and let $s_{i,j} \in \mathrm{GF}(2^q)$ be the share of b_j given to party P_i. Then, the total length of the share of each party is $\ell \cdot q = \ell \lceil \log(n+1) \rceil \leq \ell \log n + \ell$.

Clearly, every bit of the secret can be securely reconstructed. To evaluate other linear functions of the secret, we use the homomorphic property of Shamir's scheme, as observed by Benaloh [6], which enables the evaluation of linear combinations of a shared secret without revealing other information on the secret. To explain how this property is used we first observe that it is sufficient to show the result for a boolean linear function, since each coordinate of $f(s)$, where $f \in \mathcal{LIN}_\ell$, is also a linear function (we use the property that this scheme is reusable).

Let f be a boolean linear function and let B be a set of (at least) t parties with shares $\langle s_{i,j} \rangle_{P_i \in B, 1 \leq j \leq \ell}$ that wishes to compute the bit $f(s)$. By Claim 2.1, there are constants $c_1, \ldots, c_\ell$ such that $f(b_1 b_2 \ldots b_\ell) = \sum_{j=1}^{\ell} c_j b_j$. Recall that for Shamir's reconstruction procedure the secret bit b_j is computed as $b_j = \sum_{P_i \in B} \delta_i s_{i,j}$, where $\delta_i \in \mathrm{GF}(2^q)$ are appropriate coefficients. Let $x_i \triangleq \sum_{j=1}^{\ell} c_j s_{i,j}$ be a "new" share of P_i. Then, $f(s) = \sum_{j=1}^{\ell} c_j b_j = \sum_{j=1}^{\ell} c_j \sum_{P_i \in B} \delta_i s_{i,j} = \sum_{P_i \in B} \delta_i \sum_{j=1}^{\ell} c_j s_{i,j} = \sum_{P_i \in B} \delta_i x_i$.

To evaluate $f(s)$ each party $P_i \in B$ computes locally the "new" share x_i and then broadcasts it. The "new" secret $f(s)$ is just the linear combination of these "new" shares. We claim that a coalition C of size $t-1$ gains no information that is not implied by $f(s)$ during this process. Indeed, the probability of the view of C is,

$$\begin{aligned} \Pr[\langle s_{i,j} \rangle_{P_i \in C, 1 \leq j \leq \ell} \circ \langle x_i \rangle_{P_i \in B} \mid s] &= \Pr[\langle s_{i,j} \rangle_{P_i \in C, 1 \leq j \leq \ell} \circ f(s) \mid s] \\ &= \Pr[\langle s_{i,j} \rangle_{P_i \in C, 1 \leq j \leq \ell} \mid s]. \end{aligned}$$

The first equality holds because the shares of C and $f(s)$ determine the "new" shares of B and vice-versa. The second equality holds because s determines $f(s)$. Now $\Pr[\langle s_{i,j} \rangle_{P_i \in C, 1 \leq j \leq \ell} \mid s]$ is independent of s by the security requirement prior to evaluation. So, we get the security requirement after evaluation. Furthermore, if this process is repeated more than once and different linear combinations are evaluated then C will gain no information that is not implied by these linear combinations, and the parties in B can evaluate each bit of a non-boolean linear function independently. $\square$

REMARK 3.3. Observe that in the broadcast scheme there is no need for the set B to be given as input for the evaluation: each party P_i which is available just outputs an appropriate linear combination of its shares, and its identity P_i. Then, a function can be evaluated so long as there are at least t parties that agree to evaluate the function. These parties do not need to be available at the same time, or know in advance which parties would be available.

4. Schemes for Other Families of Functions. We now show how to use the basic scheme (for linear functions) to construct schemes for other families of functions. Given a family of functions, we shall construct a longer secret such that every function in the family can be evaluated as a linear function of the longer secret. Observe that any boolean function $f : \mathrm{GF}(2^\ell) \rightarrow \mathrm{GF}(2)$ can be represented as a binary vector over GF(2) of length 2^ℓ whose i-th coordinate is $f(i)$. Similarly, any function $f : \mathrm{GF}(2^\ell) \rightarrow \mathrm{GF}(2^{\ell'})$ can be represented as an array of ℓ' binary vectors of length 2^ℓ in which the j-th vector corresponds to the j-th bit function of $f(x)$. The rank of a family of functions $\mathcal{F}$ is the smallest k for which there exist boolean functions $f_1, \ldots, f_k$ such that for every function $f \in \mathcal{F}$ there exists a renaming of it f', for which each of the ℓ' vectors representing f' is a linear combination of $f_1, \ldots, f_k$. The rank of a family does not change if we add renamings of functions. So, we can assume that $\mathcal{F}$ contains all renamings of its functions.

THEOREM 4.1. *Let $\mathcal{F}$ be any family of functions.*

- *There exists a t-out-of-n interactive private channels secret sharing scheme for $\mathcal{F}$ with shares of length* $\max\{\mathrm{rank}(\mathcal{F}), \log n + 1\}$.
- *There exists a t-out-of-n non-interactive broadcast secret sharing scheme for $\mathcal{F}$ with shares of length* $\mathrm{rank}(\mathcal{F})(\log n + 1)$.

Proof. Let $f_1, f_2, \ldots, f_k$ be a basis for the vector space spanned by the functions in $\mathcal{F}$. To share a secret s the dealer generates a new secret

$$\mathrm{E}(s) = f_1(s) \circ f_2(s) \circ \ldots \circ f_k(s)$$

of length k (where $\circ$ denotes concatenation of strings). The dealer now shares the secret $\mathrm{E}(s)$ using the basic scheme (either the interactive or broadcast version, depending on the communication model). Then, $f_i(s) = e_i(\mathrm{E}(s))$ for $i = 1, 2, \ldots, k$. Let f be the function in $\mathcal{F}$ to be evaluated. Without loss of generality, we may assume that $f : \mathrm{GF}(2^\ell) \rightarrow \mathrm{GF}(2^{\ell'})$ is such that the ℓ' vectors representing it are spanned by $f_1, \ldots, f_k$. Every bit of $f(s)$ is a linear combination of $f_1, \ldots, f_k$, and therefore of the bits of $\mathrm{E}(s)$ (i.e., $e_i(\mathrm{E}(s))$). Since addition in $\mathrm{GF}(2^k)$ is bitwise, concatenation of linear functions is a linear function too. Thus, $f(s)$ can be computed by evaluating a linear function of $\mathrm{E}(s)$. Thus, Theorem 4.1 follows by Theorems 3.1 and 3.2. □

We demonstrate the applicability of the above construction by considering the family $\mathcal{ALL}$ of all possible functions of the secret (boolean and non-boolean).

COROLLARY 4.2.

- *There exists a t-out-of-n interactive private channels secret sharing scheme for $\mathcal{ALL}_\ell$ with shares of length* $\max\left\{2^\ell - 1, \log n + 1\right\}$.
- *There exists a t-out-of-n non-interactive broadcast secret sharing scheme for $\mathcal{ALL}_\ell$ with shares of length* $(2^\ell - 1)(\log n + 1)$.

Proof. By Theorem 4.1 we have to show that the rank of $\mathcal{ALL}_\ell$ is $2^\ell - 1$. Let $d = 2^\ell - 1$. Consider the d boolean functions $f_1, \ldots, f_d$, where $f_i(x) = 1 \Leftrightarrow x = i$. To evaluate a boolean function f we notice that the functions $f(x)$ and $f(x) + 1$ are renamings of each other, so we can assume that $f(0) = 0$. Hence, $f(s) = \sum_{1 \leq i \leq d} f_i(s) f(i)$, and therefore $f_1, \ldots, f_d$ form a basis for $\mathcal{ALL}_\ell$. In this case, the secret s is encoded as the vector $\mathrm{E}(s)$ of length d in which the s-th coordinate is 1 and all the other coordinates are zero (with the exception $\mathrm{E}(0) = 0$). □

Notice that the length of the secret is ℓ, while the length of the shares is $(2^\ell - 1)$. However, there are $2^{2^\ell - 1}$ different boolean functions with domain of cardinality 2^ℓ such that every function is not a renaming of another function (leaving aside non-boolean functions). Therefore, our scheme is significantly better than the naive scheme in which we share every function (up to renaming) separately. Also the length of $E(s)$ for $\mathcal{ALL}_\ell$ must be at least $\log 2^{2^\ell - 1} = 2^\ell - 1$, so the representation for $E(s)$ that we use is the best possible for the family $\mathcal{ALL}_\ell$ in this particular scheme. It remains an interesting open question whether there exists a better scheme for this family, or whether one can prove that this scheme is optimal.

5. Non-interactive Broadcast Scheme for the Bit Family. In this section we present a non-interactive broadcast scheme for the family of bit functions, whose shares are of length $O(\ell)$ (compared to $O(\ell \log n)$ of the scheme from Theorem 3.2).

THEOREM 5.1. *Let $\ell = \Omega(\log n \log\log n)$. There exists a non-interactive broadcast t-out-of-n secret sharing scheme for the family $\mathcal{BIT}_\ell$ in which the length of the secrets is ℓ and the length of the shares is $O(\ell)$.*

We first present a meta scheme (Section 5.1). Then we show a possible implementation of the meta scheme that satisfies the conditions of the Theorem.

5.1. A Meta Scheme. Let k and h be integers (to be fixed later) such that $h \geq \lceil \log(n+1) \rceil$ and the secret s has length $\ell \leq kh$. We view the secret as a binary matrix with k rows and h columns and denote its (i,j) entry by $s[i,j]$. We construct, in a way to be specified below, a new binary matrix H with $3k$ rows and h columns. Then we share every row of H using Shamir's t-out-of-n scheme over $\mathrm{GF}(2^h)$. Since the length of every row is $h \geq \lceil \log(n+1) \rceil$, Shamir's scheme is ideal and every party gets $3k$ shares of length h. The distribution stage of this scheme is illustrated in Fig. 1. When a set B of cardinality (at least) t wants to reconstruct the bit $s[i,j]$ of the secret, the parties in B reconstruct a subset $T_{i,j}$ of rows (which depends only on i, j and not on B). We will guarantee that these rows do not give any additional information on the secret.

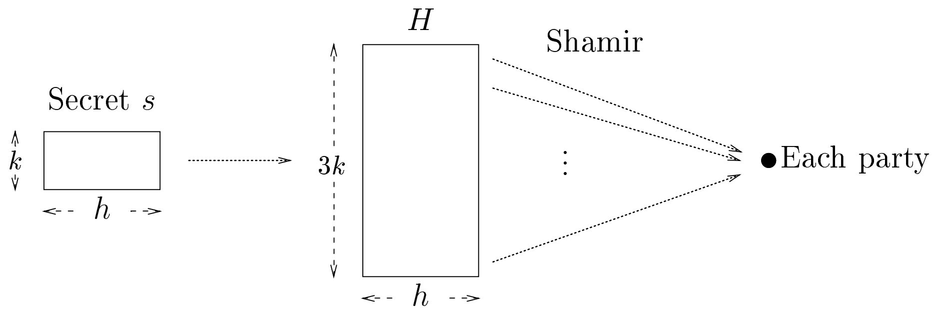

FIG. 1. *An illustration of the meta scheme of the non-interactive broadcast scheme for the bit family.*

More specifically, for every $1 \leq i \leq k$ and $1 \leq j \leq h$, we fix a set $T_{i,j} \subseteq \{1, \ldots, 3k\}$ (independently of the secret). The j-th column of H is constructed independently, and depends only on the j-th column of the secret. It is chosen uniformly at random among the column vectors such that:

$$\forall\ 1 \leq i \leq k : \sum_{a \in T_{i,j}} H[a,j] = s[i,j]\ . \tag{3}$$

To reconstruct $s[i,j]$ it is enough to reconstruct the $T_{i,j}$-rows of H, and compute the sum (modulo 2) of the j-th bit of the reconstructed rows. (Actually, every party can sum the shares of the $T_{i,j}$-rows of H, and then the parties reconstruct the secret from these shares. The j-th bit of the reconstructed secret equals to $s[i,j]$.) That is, the message of a party in a reconstructing set consists of the shares corresponding to the

$T_{i,j}$-rows of H. The existence of a matrix H satisfying Equation (3), and the security requirement after reconstruction depend on the choice of the sets $T_{i,j}$. On the other hand, independently of the choice of the sets $T_{i,j}$, any coalition of size less than t prior to any reconstruction does not have any information on the rows of H (by the properties of Shamir's scheme), and, hence, does not have any information on the secret. That is, the meta scheme is secure prior to the evaluation.

5.2. Implementing the Meta Scheme. We show how to construct sets $T_{i,j}$ such that the security requirement after reconstruction will hold. Let $R_1, R_2, \ldots, R_h \subset \{k+1, \ldots, 3k\}$ be a collection of different sets of size k. In particular, no set is contained in another set ($R_1, R_2, \ldots, R_h$ is a Sperner family [2]). For $i = 1, \ldots, k$, let $T_{i,j} \triangleq R_j \cup \{i\}$.

The number of sets of cardinality k which are contained in $\{k+1, \ldots, 3k\}$ is $\binom{2k}{k} \geq 2^{2k-o(k)} \geq 2^k$. Thus, fixing $k \triangleq \lceil \log h \rceil$ suffices. We require that $h \lceil \log h \rceil = hk \geq \ell$, and choose the smallest possible h. The length of the shares (i.e., $3h \lceil \log h \rceil$) is $\Theta(\ell)$. The following Claims are useful for proving the security of a reconstruction:

CLAIM 5.1. *For every secret s, the number of matrices H that satisfy Equation (3) for all j ($1 \leq j \leq h$) is at least 1 and is independent of s.*

Proof. Set $H[i,j] = s[i,j]$ for $1 \leq i \leq k$, and 0 otherwise. This matrix H satisfies Equation (3). To show that the number of matrices satisfying Equation (3) is independent of s, notice that H is a solution of a non-homogeneous system of linear equations. Since the system has at least one solution, the number of such matrices is the number of solutions to the homogeneous linear system of equations, where every $s[i,j] = 0$. □

To show that no coalition gains any additional information, we first consider the case in which the coalition knows only the $T_{i,j}$-rows of H. In this case, the coalition can reconstruct $s[i,j]$. We prove that the coalition does not gain any information on the other bits of the secret (i.e., bits $s[i',j']$ for $i' \neq i$ or $j' \neq j$).

CLAIM 5.2. *Let s be a secret and fix the $T_{i,j}$-rows of H such that*

$$\sum_{a \in T_{i,j}} H[a,j] = s[i,j].$$

The number of matrices H that satisfy Equation (3) for all j ($1 \leq j \leq h$) is at least 1 and is independent of s.

Proof. We first prove that there exists a matrix H satisfying the requirements. Since the columns of H are constructed independently, it is enough to prove the existence of every column j'. Only the j'-th column of the $T_{i,j}$-rows of H, and the j'-th column of s influence the j'-th column of H, hence, while considering the j'-th column we can ignore the rest of the columns.

We first consider the j-th column of H (i.e., $j' = j$). Since $T_{i',j} = R_j \cup \{i'\}$, we can assign $H[i',j]$ a value as follows: for every i' such that $k+1 \leq i' \leq 3k$ and $i' \notin R_j$ set $H[i',j] = 0$ (this is arbitrary); for every i' such that $1 \leq i' \leq k$ set $H[i',j] = \sum_{a \in R_j} H[a,j] + s[i',j]$.

The case $j' \neq j$ is similar except that we have to be careful about $s[i,j']$. Since $R_{j'} \not\subseteq R_j$, there is an element $p \in T_{i,j'} \setminus T_{i,j}$. We can assign $H[i',j']$ a value as follows: for every i' such that $k+1 \leq i' \leq 3k$, $i' \notin R_j$, and $i' \neq p$ set $H[i',j'] = 0$; $H[p,j'] = \sum_{a \in T_{i,j'} \setminus \{p\}} H[a,j'] + s[i,j']$. For every i', $1 \leq i' \leq k$, and $i' \neq i$ set $H[i',j'] = \sum_{a \in R_{j'}} H[a,j'] + s[i',j']$.

We have shown that there exists a matrix as required. By the same arguments as in the proof of Claim 5.1, the number of possible matrices given a secret s, is independent of s. □

We are now ready to prove the security requirement (note that this scheme is *not* reusable).

CLAIM 5.3. *The above implementation of the meta scheme is secure.*

Proof. Let B be any set that evaluates a bit $s[i,j]$ of the secret, and C be any coalition. The security requirement before reconstruction is easy (this was discussed at the end of Section 5.1). Suppose that C has obtained the messages sent by the parties in B. That is, the parties in C know the shares of the parties of B corresponding to the $T_{i,j}$-rows of H. Therefore, the only information that they gain is the $T_{i,j}$-rows of H. Then, given the shares of the coalition and all the messages that were sent, every matrix H that agrees with these rows is possible. We need to prove that, given two secrets in which the (i,j)-th bit is the same, and a matrix H in which only the $T_{i,j}$-rows are fixed: the probability that H was constructed from each secret is equally likely. By Claim 5.1 and Claim 5.2 this probability only depends on $s[i,j]$. □

6. Characterization of Families with Ideal Schemes. In this section we consider *ideal* secret sharing schemes for a family $\mathcal{F}$. We say that two secrets s, s' are *distinguishable* by $\mathcal{F}$ if there exists a function $f \in \mathcal{F}$ for which $f(s) \neq f(s')$. A secret sharing scheme for $\mathcal{F}$ is *ideal* if the cardinality of the domain of shares equals the cardinality of the domain of distinguishable secrets. These are the shortest shares possible by [26]. The following theorem gives several impossibility results for ideal schemes for a family $\mathcal{F}$ which contains the bit functions.

THEOREM 6.1 (Characterization Theorem).

(1) *Let $2 \leq t \leq n$. Suppose that there is an ideal* interactive *t-out-of-n secret sharing scheme for a family of functions $\mathcal{F}$ such that $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. Then, any* boolean *function $f \in \mathcal{F}$ is linear.*

(2) *Let $t \in \{2, n\}$. Suppose that there is an ideal* non-interactive *t-out-of-n secret sharing scheme for a family of functions $\mathcal{F}$ such that $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. Then, $\mathcal{F} \subseteq \mathcal{LIN}_\ell$.*

(3) *Let $3 \leq t \leq n-1$. In an ideal t-out-of-n secret sharing the evaluation of every non-constant boolean function requires interaction on private channels.*

(4) *Let $t = 2$. In an ideal non-interactive 2-out-of-n secret sharing every non-constant boolean function cannot be evaluated via a broadcast channel.*

It follows that for families $\mathcal{F}$, with $\mathcal{BIT}_\ell \subseteq \mathcal{F}$, if interaction is allowed then every boolean function in $\mathcal{F}$ is linear. If interaction is not allowed, then either $t = 2$ or $t = n$, and all the functions in $\mathcal{F}$ are linear. In particular when $t = 2$ private channels must be used.

In the Section 6.1 we will discuss some basic properties of ideal schemes that will be needed for the proof of this Theorem. In Section 6.2 we prove (1) and (2), and in Section 6.3 we prove (3) and (4).

6.1. Properties of Ideal Schemes. In this section we discuss some simple properties of ideal schemes which are useful in the sequel.

PROPOSITION 6.2 ([26]). *Fix the shares of any $t-1$ parties in an ideal t-out-of-n secret sharing scheme. Then, the share of any other party P is a permutation of the secret. In particular all shares are possible for P.*

In our proof, we use the following proposition regarding private functions, that is functions that can be computed without revealing any other information on the inputs.

(For formal definition of privacy the reader is referred to Appendix C.) Bivariate boolean private functions were characterized by Chor and Kushilevitz [16]:

PROPOSITION 6.3 ([16]). *Let A_1, A_2 be nonempty sets and $f : A_1 \times A_2 \to \{0,1\}$ be an arbitrary boolean function. Then, f can be computed privately if and only if there exist boolean functions $f_1 : A_1 \to \{0,1\}$, $f_2 : A_2 \to \{0,1\}$ such that for every $x \in A_1$, $y \in A_2$ it holds that $f(x,y) = f_1(x) + f_2(y)$.*

CLAIM 6.1. *Let f be a boolean function that can be evaluated securely in an ideal 2-out-of-2 secret sharing scheme. Then, there exist boolean functions f_1 and f_2 such that $f(s) = f_1(x) + f_2(y)$, where x and y are the shares of P_1 and P_2 respectively.*

Proof. The function f is evaluated from x and y, i.e., there is some function f' such that $f(s) = f'(x,y)$. By Proposition 6.2 any information that a party gets on the share of the other party is translated to information on the secret. That is, the parties must compute $f'(x,y)$ in such a way that each party receives only information implied by $f'(x,y)$. In other words, they compute the boolean function f' privately. By Proposition 6.3 there exist f_1 and f_2 such that $f(s) = f'(x,y) = f_1(x) + f_2(y)$. □

Consider an ideal t-out-of-n secret sharing scheme for $\mathcal{F}$. Without loss of generality, we may assume that all the secrets are distinguishable by $\mathcal{F}$. So, the parties $P_1, P_2, \ldots, P_t$ can reconstruct the secret from their shares. Denote the reconstructing function by $h(s_1, s_2, \ldots, s_t)$. We next prove an important property of the messages sent in a non-interactive protocol that evaluates f.

CLAIM 6.2. *Let f be a function that can be securely evaluated in an ideal t-out-of-n secret sharing scheme without interaction. Without loss of generality, we may assume that all the messages sent by P_i, while holding the share s_i, to the other parties are identical, and equal to*

$$f(h(\underbrace{0,\ldots,0}_{i-1}, s_i, \underbrace{0,\ldots,0}_{t-i})).$$

Proof. It is sufficient to prove the claim for P_1. Suppose that the shares of the parties $P_2, \ldots, P_t$ are $s_i = 0$. Consider a possible message of P_1 while the set $\{P_1, P_2, \ldots, P_t\}$ computes the value $f(s)$. Party P_1 might toss coins, so if there are several messages choose the first one lexicographically. When P_1 holds two shares s_1 and s_1' such that $f(h(s_1, 0, \ldots, 0)) \neq f(h(s_1', 0, \ldots, 0))$, the messages of P_1 to P_i have to be different, otherwise P_i will compute an incorrect value in one of the cases. On the other hand, for two shares s_1 and s_1' such that $f(h(s_1, 0, \ldots, 0)) = f(h(s_1', 0, \ldots, 0))$ the messages sent by P_1 to the parties in $\{P_2, \ldots, P_t\}$ have to be the same in both cases, otherwise the coalition $\{P_2, \ldots, P_t\}$ can distinguish between the secrets $s = h(s_1, 0, \ldots, 0)$ and $s' = h(s_1', 0, \ldots, 0)$ although $f(s) = f(s')$. Thus, without loss of generality, the messages sent by P_1 while holding the share s_1 are identical and equal to $f(h(s_1, 0, \ldots, 0))$. □

6.2. Proofs of (1) and (2) of Theorem 6.1. The proofs of Items (1) and (2) are similar, and have two stages. We consider the following two-party distribution scheme over $\mathrm{GF}(2^\ell)$, denoted XOR. Given a secret s, the dealer chooses at random $x \in \mathrm{GF}(2^\ell)$. The share of the first party is x, and the share of the second party is $y = s + x$. In the first stage (Section 6.2.1) we characterize the functions that can be evaluated from the shares of XOR. In the second stage (Section 6.2.2) we show that if there exists an ideal t-out-of-n secret sharing scheme for $\mathcal{F}$, then XOR is a secret sharing scheme for $\mathcal{F}$. The combination of the two stages implies Items (1) and (2).

6.2.1. Characterizing the Functions which can be Evaluated with XOR. We first characterize the functions that can be evaluated with XOR *without interaction*. It can be seen that every linear function of the secret can be evaluated with XOR (the details are the same as in Remark 3.1). We prove that no other functions can be evaluated securely without interaction. Then we characterize the boolean functions that can be evaluated with XOR *with interaction* and show that these functions are exactly the boolean linear functions.

To prove the characterization without interaction, we will prove that $f(s)$ can be computed from $f(x)$ and $f(y)$ (where x and y are the shares). In the next claim we prove that every function with this property is a renaming of a linear function.

CLAIM 6.3. *Let $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ be any function. If there exists a function g such that $f(x+y) = g(f(x), f(y))$ for every $x, y \in \mathrm{GF}(2^\ell)$, then f is a renaming of a linear function.*

Proof. Let $X_0 \triangleq \{x : f(x) = f(0)\}$. We first prove that

$$f(x_1) = f(x_2) \text{ if and only if } f(x_1 + x_2) = f(0) \ . \tag{4}$$

The "only if" direction follows from: $f(x_1+x_2) = g(f(x_1), f(x_2)) = g(f(x_1), f(x_1)) = f(x_1 + x_1) = f(0)$. Similarly the "if" direction follows from the following simple equations:

$$f(x_1) = f(x_1+0) = g(f(x_1), f(0)) = g(f(x_1), f(x_1+x_2)) = f(x_1+x_1+x_2) = f(x_2) \ .$$

That is, if $x_1, x_2 \in X_0$, then $x_1 + x_2 \in X_0$. In other words the set X_0 is a linear space over the field GF(2). Now consider a linear transformation $f' : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ whose null space is X_0 (that is, $f'(x) = 0$ if and only if $x \in X_0$). Since X_0 is a linear space, such linear transformations exists. We claim that f is a renaming of f', i.e., $f(x) = f(y)$ if and only if $f'(x) = f'(y)$. By Equation (4), $f(x) = f(y)$ if and only if $x + y \in X_0$. Furthermore, $x + y \in X_0$ if and only if $f'(x + y) = 0$ (by the definition of f'). Since f' is linear, $f'(x) + f'(y) = f'(x + y)$. Hence, $f(x) = f(y)$ if and only if $f'(x) = f'(y)$. □

CLAIM 6.4. *Let f be any function that can be evaluated without interaction with the scheme* XOR. *Then, f is a renaming of a linear function.*

Proof. Assume, without loss of generality, that $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ (otherwise consider the function of x defined as $\min\{y : f(y) = f(x)\}$, which is a renaming of f). By Claim 6.3 it suffices to prove that $f(x + y)$ can be computed from $f(x)$ and $f(y)$.

Denote the share of P_1 by x and the share of P_2 by y. Recall that $s = x + y$. By Claim 6.2 the message sent by P_2 while holding a share y to P_1 is $f(y)$, and the message sent by P_1 while holding a share x to P_2 is $f(x)$. Hence, P_1 can compute $f(s) = f(x+y)$ from x and $f(y)$. Moreover, for every two shares x_1 and x_2 held by P_1 such that $f(x_1) = f(x_2)$ and every share y held by P_2, party P_1 must compute the same value of $f(s)$, since P_2 receives the same message from P_1 in both cases, and therefore computes the same value of $f(s)$. Hence, P_1 can compute $f(x + y)$ from $f(x)$ and $f(y)$. The computed function is the desired function g with $f(x+y) = g(f(x), f(y))$. By Claim 6.3 this implies that f is a renaming of a linear function. □

We now prove a similar claim for interactive evaluations. However, this claim applies only to boolean functions.

CLAIM 6.5. *Let f be a* boolean *function that can be evaluated interactively with the scheme* XOR. *Then, f is a renaming of a linear function.*

Proof. Since XOR is an ideal scheme, Claim 6.1 implies that there exists boolean functions f_1, f_2 such that $f(s) = f(x+y) = f_1(x) + f_2(y)$. In particular $f(x) = f(x+$

$0) = f_1(x) + f_2(0)$. Similarly $f(y) = f_1(0) + f_2(y)$. So, there exists a bit b such that $f(x+y) = f(x) + f(y) + b$. Then, $f'(x) = f(x) + b$ is a linear function which is a renaming of f. □

The exact family of functions which can be evaluated interactively with XOR consists of the two-argument functions for which $f(s) = f(x+y)$ can be computed privately from x and y (as characterized in [29]).

6.2.2. Reduction to XOR. In this section we prove that if there exists an ideal t-out-of-n secret sharing scheme for a family $\mathcal{F}$, then there exists an ideal 2-out-of-2 secret sharing scheme for $\mathcal{F}$. We then prove that this implies that XOR is a secret sharing scheme for $\mathcal{F}$.

CLAIM 6.6. *If there exists an interactive (respectively non-interactive) ideal t-out-of-n secret sharing scheme for $\mathcal{F}$, then there exists an interactive (respectively non-interactive) ideal* 2*-out-of-*2 *secret sharing scheme for $\mathcal{F}$.*

Proof. Let $B = \{P_1, P_2, \ldots, P_t\}$ be a set of parties. We ignore the shares distributed to parties not in B. Therefore, we have an ideal t-out-of-t secret sharing scheme for $\mathcal{F}$. Let $\langle s_1, s_2, s_3, \ldots, s_t \rangle$ be any (fixed) vector of shares that is dealt to B with positive probability. To share a secret s the dealer now generates a random vector of shares of s (according to the scheme) that agrees with $s_3, \ldots, s_t$ respectively, and gives the first two components of this vector to P_1 and P_2. Parties P_1 and P_2 know the shares of the other parties (as they are fixed) and therefore P_1, say, can simulate the parties $P_3, \ldots, P_t$ in the protocol which evaluates the function $f \in \mathcal{F}$. Hence, P_1 and P_2 can evaluate $f(s)$ from their shares and the messages they exchange. On the other hand, P_1 has no more information than the information known to the coalition $\{P_1, P_3, \ldots, P_t\}$ of cardinality $t-1$ in the t-out-of-t scheme. So, P_1 does not gain additional information on s. Similar arguments hold for P_2. This implies that this scheme is a 2-out-of-2 secret sharing scheme for $\mathcal{F}$. □

CLAIM 6.7. *Let $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. If there exists an interactive (respectively non-interactive) ideal t-out-of-n secret sharing scheme for $\mathcal{F}$, then* XOR *is an interactive (respectively non-interactive) secret sharing scheme for $\mathcal{F}$.*

Proof. By Claim 6.6 we can assume that there exists an ideal 2-out-of-2 secret sharing scheme for $\mathcal{F}$, say Π. We transform Π into XOR in a way that every function that can be evaluated in Π can also be evaluated in XOR.

Claim 6.1 implies that for every j, where $1 \le j \le \ell$, there exist boolean functions f_1^j, f_2^j such that $e_j(s) = f_1^j(x) + f_2^j(y)$. For $i = 1, 2$, define $\overline{\mathrm{m}}_i(x)$ as the concatenation of the values of these functions, that is $\overline{\mathrm{m}}_i(x) \triangleq f_i^1(x) \circ f_i^2(x) \circ \ldots \circ f_i^\ell(x)$. Then, for every secret s and random input r of the dealer $s = \overline{\mathrm{m}}_1(\Pi_1(s,r)) + \overline{\mathrm{m}}_2(\Pi_2(s,r))$. Therefore, the scheme Π' defined by $\Pi'(s,r) = \langle \overline{\mathrm{m}}_1(\Pi_1(s,r)), \overline{\mathrm{m}}_2(\Pi_2(s,r)) \rangle$ is a distribution scheme equivalent to XOR. We still have to show that the two parties can evaluate securely every function in $\mathcal{F}$ with Π'.

We first prove that $\overline{\mathrm{m}}_1$ is invertible. Clearly, the two parties can reconstruct the secret in Π', while every single party knows nothing about the secret. Therefore, Π' is a 2-out-of-2 secret sharing scheme and, by [26], the cardinality of the domain of shares is at least the cardinality of the domain of secrets. Thus, the cardinality of the range of $\overline{\mathrm{m}}_1$ is at least as large as 2^ℓ. The domain of $\overline{\mathrm{m}}_1$, which is the set of shares of P_1, has cardinality 2^ℓ. Therefore, $\overline{\mathrm{m}}_1$ is a bijection and, hence, invertible. So, P_1 can reconstruct the share x from $\overline{\mathrm{m}}_1(x)$. Similarly, P_2 can reconstruct the share y from $\overline{\mathrm{m}}_2(y)$. This implies that the parties P_1 and P_2, while holding $\overline{\mathrm{m}}_1(x)$ and $\overline{\mathrm{m}}_2(y)$ respectively, can evaluate every function $f \in \mathcal{F}$. We have proved that every function $f \in \mathcal{F}$ can be evaluated with Π' which is equivalent to XOR. Furthermore, if the evaluation of the original scheme required no interaction, then also the evaluation with the XOR scheme requires no interaction. □

6.3. Proofs of (3) and (4) of Theorem 6.1. Next we prove Item (4). Namely, in any ideal non-interactive t-out-of-n secret sharing scheme ($2 \leq t \leq n-1$) evaluation of a boolean function requires private channels. The proof is by contradiction. Suppose that there is an ideal non-interactive t-out-of-n scheme in which the evaluation can be held on a broadcast channel. Then, by the same arguments as in Claim 6.6, since $t \leq n-1$, there is an ideal 2-out-of-3 scheme with the same property. Our first claim is that, without loss of generality, in the evaluation of $f(s)$ every party sends a one bit message such that the sum of the messages is $f(s)$.

CLAIM 6.8. *Assume there exists an ideal* 2*-out-of-*3 *secret sharing scheme in which a non-constant boolean function f can be securely evaluated without interaction via a broadcast channel. Then, P_1 and P_2 can securely compute $f(s)$ by sending one bit messages m_1 and m_2, such that $m_1 + m_2 = f(s)$.*

Proof. By Claim 6.2 we can assume that the message of P_1 while holding the share s_1 is the bit $m_1 = f(h(s_1, 0))$. Similarly, we assume that the message of P_2 while holding the share s_2 is $m_2 = f(h(0, s_2))$. Furthermore, the value of $f(s)$ is determined by these two messages. There are two values for $f(s)$ and two values for each message. Since each party knows nothing about $f(s)$, a change of any message will result in a change of $f(s)$. The only boolean functions with two binary variables satisfying this requirement are addition modulo 2, i.e., $m_1 + m_2 + b$ with either $b = 0$ or $b = 1$. So, without loss of generality, assume that P_1 sends the message $m_1 + b$ and the claim follows. □

CLAIM 6.9. *Consider an ideal non-interactive* 2*-out-of-*3 *secret sharing scheme for a family $\mathcal{F}$ which contains a non-constant boolean function f. Then, f cannot be evaluated via a broadcast channel.*

Proof. Fix the share of P_3 to be $z = 0$. By Proposition 6.2, the share x of P_1 is a permutation of the secret. Assume, without loss of generality, that this share is equal to the secret. (Of course P_1 does not know that $z = 0$ and does not know the secret.) Since P_3 does not know $f(s)$ although $z = 0$, the value of $f(s)$ is not fixed, and the message that P_1 sends to P_2 while P_1 and P_2 evaluate $f(s)$ cannot be constant (otherwise P_2 will not be able to evaluate $f(s)$). Furthermore, for any two shares x, x' held by P_1 such that $f(x) = f(x')$, the messages of P_1 to P_2 should be the same (otherwise P_3 will distinguish between the two secrets). Without loss of generality, the message m_1 which P_1 sends to P_2 is $f(x)$. By Claim 6.8, $f(s) = m_1 + m_2$. Thus, the message that P_2 sends to P_1 is constant and P_1 will not be able to evaluate $f(s)$, which leads to a contradiction. Hence, $f(s)$ cannot be evaluated on a broadcast channel. □

We conclude this section by proving Item (3). That is, for $3 \leq t \leq n-1$, the evaluation requires interaction via private channels. Again, we assume towards contradiction that there exists an ideal non-interactive t-out-of-n scheme for the family $\mathcal{F}$, and construct an ideal non-interactive 3-out-of-4 scheme for the family $\mathcal{F}$. We show that this contradicts Claim 6.9.

CLAIM 6.10. *Assume there is an ideal* 3*-out-of-*4 *secret sharing scheme for a family $\mathcal{F}$ in which a function $f \in \mathcal{F}$ can be evaluated via private channels without interaction. Then, there is an ideal* 2*-out-of-*3 *secret sharing scheme in which f can be evaluated via a broadcast channel.*

Proof. By Claim 6.2 we may assume that the messages sent during the evaluation by one party to the other two parties are identical. In particular, if $\{P_i, P_j, P_k\}$

evaluates a function, then P_i knows the messages that P_j, P_k exchange. To construct a 2-out-of-3 secret sharing scheme, we fix a share u for P_4, and share the secret s with a random vector of shares such that the share of P_4 is u. When P_i and Pj want to evaluate $f(s)$, where $1 \le i < j \le 3$, then P_i simulates P_4, and all messages and broadcasted. After this evaluation, every party P_k, where $k \in \{1, 2, 3\}$, will have the same information as the coalition $\{P_k, P_4\}$ had in the original scheme. Hence, P_k does not gain any information that is not implied by $f(s)$. That is, we get a 2-out-of-3 secret sharing scheme for $\mathcal{F}$. $\square$

7. Discussion. In this section we consider possible extensions of our work. So far, we considered only *threshold* secret sharing schemes. In [22] a general notion of secret sharing schemes for arbitrary collections of reconstructing sets is defined. That is, we are given a collection $\mathcal{A}$ of sets of parties called an *access structure* and require that any set in $\mathcal{A}$ can reconstruct the secret, while any set not in $\mathcal{A}$ does not know anything about the secret. Clearly, secret sharing schemes exist only for monotone collections. Conversely, it is known that for every monotone collection there exists a secret sharing scheme [22]. More efficient schemes for general access structures were presented (e.g., in [7, 36]). However, the length of the shares in these schemes can be exponential in the number of parties (i.e., of length $\ell 2^{\Theta(n)}$ where n is the number of parties in the system and ℓ is the length of the secret). Our definition of secret sharing for a family $\mathcal{F}$ of functions can be naturally generalized for an arbitrary access structure. To construct such schemes, observe that the schemes of [22, 7, 36] are "linear": the share of each party is a vector of elements over some field, and every set in $\mathcal{A}$ reconstructs the secret using a linear combination of elements in their shares. Thus, if we share every bit of the secret independently, we can evaluate every linear function of the secret without any interaction (the details are as in Theorem 3.2). This implies that for every access structure, there exists a scheme for the family $\mathcal{ALL}_\ell$ in which the length of the shares is $O(2^\ell 2^n)$. However, if the access structure has a more efficient linear scheme for sharing a single bit then the length of the shares can be shorter (but at least 2^ℓ).

Another possible extension of the definition is by allowing a weaker "non-perfect" definition of security. Such extensions in the context of traditional secret sharing schemes were considered, e.g., in [9]. As remarked, there are several related works on *simultaneous* secret sharing. For example, in [23] each set of cardinality at least t should be able to reconstruct some of the secrets (but not all). In [11] each set B of cardinality at least t is able to reconstruct any of the distributed secrets.[6] The security requirement considered in [11, 24, 10] is somewhat weak: after revealing one of the secrets, limited information about other secrets may be leaked.

Our scheme for the family of functions $\mathcal{ALL}_\ell$ uses shares of size 2^ℓ. We can construct an efficient scheme if we relax the security requirement as follows: every set of size at least t can evaluate any function $f(s)$ in a way that every coalition of at most $\lfloor (t-1)/2 \rfloor$ parties gains no information on the secret (but larger coalitions may get some information). In this case, we can use Shamir's t-out-of-n scheme to distribute the secret and the interactive protocol of [5] or [15] to securely compute $f(s)$. In this scenario, the length of the share is the same as the length of the secret. However, for most functions f, the length of the communication in the evaluation of f is exponential in ℓ.

[6] In fact [11] deals with more general *access structures* and not only threshold schemes.

Acknowledgment. The authors would like to thank the referees for several helpful comments.

REFERENCES

[1] M. Abadi, J. Feigenbaum, and J. Kilian, *On hiding information from an oracle*, J. of Computer and System Sciences, 39 (1989), pp. 21–50.

[2] M. Aigner, *Combinatorial Theory*, Springer-Verlag, Berlin, 1979.

[3] D. Beaver and J. Feigenbaum, *Hiding instances in multioracle queries*, in STACS '90, 7th Annu. Symp. on Theoretical Aspects of Computer Science, C. Choffrut and T. Lengauer, eds., vol. 415 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 37–48.

[4] A. Beimel and B. Chor, *Universally ideal secret sharing schemes*, IEEE Trans. on Information Theory, 40 (1994), pp. 786–794.

[5] M. Ben-Or, S. Goldwasser, and A. Wigderson, *Completeness theorems for noncryptographic fault-tolerant distributed computations*, in Proc. of the 20th Annu. ACM Symp. on the Theory of Computing, 1988, pp. 1–10.

[6] J. Benaloh, *Secret sharing homomorphisms: Keeping shares of a secret secret*, in Advances in Cryptology - CRYPTO '86, A. M. Odlyzko, ed., vol. 263 of Lecture Notes in Computer Science, Springer-Verlag, 1987, pp. 251–260.

[7] J. Benaloh and J. Leichter, *Generalized secret sharing and monotone functions*, in Advances in Cryptology - CRYPTO '88, S. Goldwasser, ed., vol. 403 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 27–35.

[8] G. R. Blakley, *Safeguarding cryptographic keys*, in Proc. of the 1979 AFIPS National Computer Conference, R. E. Merwin, J. T. Zanca, and M. Smith, eds., vol. 48 of AFIPS Conference proceedings, AFIPS Press, 1979, pp. 313–317.

[9] G. R. Blakley and C. Meadows, *The security of ramp schemes*, in Advances in Cryptology - CRYPTO '84, G. R. Blakley and D. Chaum, eds., vol. 196 of Lecture Notes in Computer Science, Springer-Verlag, 1985, pp. 242–268.

[10] C. Blundo, A. De Santis, G. Di Crescenzo, A. Giorgio Gaggia, and U. Vaccaro, *Multi-secret sharing schemes*, in Advances in Cryptology - CRYPTO '94, Y. Desmedt, ed., vol. 839 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 150–163.

[11] C. Blundo, A. De Santis, and U. Vaccaro, *Efficient sharing of many secrets*, in STACS '93, P. Enjalbert, A. Finkel, and K. W. Wagner, eds., vol. 665 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 692–703.

[12] E. F. Brickell, *Some ideal secret sharing schemes*, Journal of Combin. Math. and Combin. Comput., 6 (1989), pp. 105–113.

[13] E. F. Brickell and D. M. Davenport, *On the classification of ideal secret sharing schemes*, J. of Cryptology, 4 (1991), pp. 123–134.

[14] E. F. Brickell and D. R. Stinson, *Some improved bounds on the information rate of perfect secret sharing schemes*, J. of Cryptology, 5 (1992), pp. 153–166.

[15] D. Chaum, C. Crépeau, and I. Damgard, *Multiparty unconditionally secure protocols*, in Proc. of the 20th Annu. ACM Symp. on the Theory of Computing, 1988, pp. 11–19.

[16] B. Chor and E. Kushilevitz, *A zero-one law for Boolean privacy*, SIAM J. on Discrete Mathematics, 4 (1991), pp. 36–47.

[17] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, *How to share a function securely*, in Proc. of the 26th Annu. ACM Symp. on the Theory of Computing, 1994, pp. 522–533.

[18] Y. Desmedt, *Threshold cryptosystems*, in Advances in Cryptology - AUSCRYPT '92, J. Seberry and Y. Zheng, eds., vol. 718 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 5–14.

[19] Y. Desmedt and Y. Frankel, *Shared generation of authenticators and signatures*, in Advances in Cryptology - CRYPTO '91, J. Feigenbaum, ed., vol. 576 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 457–469.

[20] ———, *Homomorphic zero-knowledge threshold schemes over any finite abelian group*, SIAM J. on Discrete Mathematics, 7 (1994), pp. 667–679.

[21] M. K. Franklin and M. Yung, *Communication complexity of secure computation*, in Proc. of the 24th Annu. ACM Symp. on the Theory of Computing, 1992, pp. 699–710.

[22] M. Ito, A. Saito, and 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. of Cryptology, 6 (1993), pp. 15–20.

[23] W. Jackson, K. M. Martin, and C. M. O'Keefe, *Multisecret threshold schemes*, in Advances

in Cryptology - CRYPTO '93, D. R. Stinson, ed., vol. 773 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 126–135.

[24] ———, *On sharing many secrets*, in ASIACRYPT '94, J. Pieprzyk and R. Safavi-Naini, eds., vol. 917 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 42–54.

[25] ———, *Ideal secret sharing schemes with multiple secrets*, J. of Cryptology, 9 (1996), pp. 233–250.

[26] E. D. KARNIN, J. W. GREENE, AND M. E. HELLMAN, *On secret sharing systems*, IEEE Trans. on Information Theory, 29 (1983), pp. 35–41.

[27] J. KILIAN AND N. NISAN, *Private communication*, 1990.

[28] S. C. KOTHARI, *Generalized linear threshold scheme*, in Advances in Cryptology - CRYPTO '84, G. R. Blakley and D. Chaum, eds., vol. 196 of Lecture Notes in Computer Science, Springer-Verlag, 1985, pp. 231–241.

[29] E. KUSHILEVITZ, *Privacy and communication complexity*, SIAM J. on Discrete Mathematics, 5 (1992), pp. 273–284.

[30] R. J. MCELIECE AND D. V. SARWATE, *On sharing secrets and Reed-Solomon codes*, Communications of the ACM, 24 (1981), pp. 583–584.

[31] M. O. RABIN, *Randomized Byzantine generals*, in Proc. of the 24th Annu. IEEE Symp. on Foundations of Computer Science, 1983, pp. 403–409.

[32] R. L. RIVEST, L. ADLEMAN, AND M. L. DERTOUZOS, *On data banks and privacy homomorphisms*, in Foundations of Secure Computation, R. A. DeMillo, D. P. Dobkin, A. K. Jones, and R. J. Lipton, eds., Academic Press, 1978, pp. 169–179.

[33] P. D. SEYMOUR, *On secret-sharing matroids*, J. of Combinatorial Theory, Series B, 56 (1992), pp. 69–73.

[34] A. SHAMIR, *How to share a secret*, Communications of the ACM, 22 (1979), pp. 612–613.

[35] G. J. SIMMONS, *An introduction to shared secret and/or shared control and their application*, in Contemporary Cryptology, The Science of Information Integrity, G. J. Simmons, ed., IEEE Press, 1992, pp. 441–497.

[36] G. J. SIMMONS, W. JACKSON, AND K. M. MARTIN, *The geometry of shared secret schemes*, Bulletin of the ICA, 1 (1991), pp. 71–88.

[37] D. R. STINSON, *An explication of secret sharing schemes*, Designs, Codes, and Cryptography, 2 (1992), pp. 357–390.

Appendix A. Equivalence of Definitions of Security. An alternative definition to our definition of security of secret sharing for $\mathcal{F}$ requires that after evaluating $f(s)$, the probability of the secret given the view of the coalition is the same as the probability of the secret given only the value $f(s)$. We prove that these two definitions are equivalent with respect to private channels schemes.

CLAIM A.1. *Let X be any random variable on the set of secrets S such that $\Pr[\,X = s\,] > 0$ for every s. A private channels secret sharing scheme for the family $\mathcal{F}$ is secure (according to Definition 2.5) if and only if, for any value v in the range of f, any secret $s \in S$ such that $f(s) = v$, any coalition C of size less than t such that $B \cap C \neq \emptyset$, and any value* view *of $VIEW_C$ such that* $\Pr[\,VIEW_C = \text{view}\,] > 0$,

$$\Pr[\,X = s \mid VIEW_C = \text{view}\,] \quad = \quad \Pr[\,X = s \mid f(X) = v\,]\,, \tag{5}$$

where the probability is taken over the distribution of the secrets, the random input of the dealer and the random inputs of the parties.[7]

Proof. Assume that the scheme is secure according to Definition 2.5, that is for every $s, s' \in S$ such that $f(s) = f(s')$: $\Pr[\mathrm{VIEW}_C = \text{view} \mid X = s] = \Pr[\mathrm{VIEW}_C = \text{view} \mid X = s']$. Then, $\Pr[\mathrm{VIEW}_C = \text{view} \mid X = s] = \Pr[\mathrm{VIEW}_C = \text{view} \mid f(X) = v]$. Using Bayes' rule and the previous equation we get,

$$\Pr[X = s \mid \mathrm{VIEW}_C = \text{view}]$$

[7]For broadcast schemes we have to deal also with coalitions C such that $B \cap C = \emptyset$. For these coalitions (5) does not necessarily hold as they might not learn the value $f(s)$ (but will learn some information on $f(s)$).

$$
\begin{aligned}
&= \frac{\Pr[\mathrm{VIEW}_C = \mathrm{view} \mid X = s] \cdot \Pr[X = s]}{\Pr[\mathrm{VIEW}_C = \mathrm{view}]} \\
&= \frac{\Pr[\mathrm{VIEW}_C = \mathrm{view} \mid f(X) = v] \cdot \Pr[X = s]}{\Pr[\mathrm{VIEW}_C = \mathrm{view}]} \\
&= \frac{\Pr[f(X) = v \mid \mathrm{VIEW}_C = \mathrm{view}] \cdot \Pr[X = s]}{\Pr[f(X) = v]} \\
&= \Pr[f(X) = v \mid \mathrm{VIEW}_C = \mathrm{view}] \cdot \Pr[X = s \mid f(X) = v] \ .
\end{aligned}
$$

For the last equality recall that $f(s) = v$, i.e., $\Pr[X = s] = \Pr[(X = s) \circ (f(X) = v)]$. Furthermore, each party in B can evaluate $f(X)$ from the information it has. Since $B \cap C \neq \emptyset$, the probability that $\Pr[f(X) = v \mid \mathrm{VIEW}_C = \mathrm{view}]$ is 1. Hence, the security according to Definition 2.5, implies Equation (5).

The proof in the other direction is similar. Assume that Equation (5) holds. Using Bayes' rule we see that for every s such that $f(s) = v$,

$$
\begin{aligned}
\Pr[\mathrm{VIEW}_C = \mathrm{view} \mid X = s] &= \frac{\Pr[f(X) = v \mid X = s] \cdot \Pr[\mathrm{VIEW}_C = \mathrm{view}]}{\Pr[f(X) = v]} \\
&= \frac{\Pr[\mathrm{VIEW}_C = \mathrm{view}]}{\Pr[f(X) = v]} \ .
\end{aligned}
$$

For the last equality notice that $\Pr[f(X) = v \mid X = s] = 1$ (since $f(s) = v$). Thus, $\Pr[\mathrm{VIEW}_C = \mathrm{view} \mid X = s]$ is independent of s, which implies the security of the scheme. □

Appendix B. Shamir's t-out-of-n Threshold Scheme. For the sake of completeness we describe the t-out-of-n secret sharing scheme of Shamir [34]. In this scheme, the set of secrets is identified with the elements of the field $\mathrm{GF}(q)$, where q is a prime power larger than n, the number of parties in the system. To share a secret $s \in \mathrm{GF}(q)$ the dealer chooses independently with uniform distribution $t-1$ random elements $r_1, \ldots, r_{t-1}$ in $\mathrm{GF}(q)$. These elements and the secret s define the polynomial $p(x) = r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \ldots + r_1 x + s$, for which $p(0) = s$. The dealer gives to each party P_j the share $s_j = p(j)$. Any set B of at least t parties can now reconstruct $p(x)$ by interpolation,

$$
s = p(0) = \sum_{P_j \in B} s_j \prod_{P_i \in B;\, i \neq j} \frac{i}{i-j} \ .
$$

Observe that the secret s is a linear combination of the shares s_j of the parties $P_j \in B$, for which the coefficient of s_j is

$$
\delta_{B,j} \triangleq \prod_{P_i \in B;\, i \neq j} \frac{i}{i-j} \ .
$$

Appendix C. Definition of Private Computations. We define *private protocols*, which are used for evaluating functions of the secrets securely. The model of computation is the private channels model described in Definition 2.2 and Definition 2.3. We say that a coalition C *gains no additional information* from the execution of a randomized protocol F which computes f, if the view of C does not imply any information other than what follows from its input and the function value. That is, for every two input vectors $\langle x_i \rangle, \langle x'_i \rangle$ of F for which $\langle x_i \rangle_C = \langle x'_i \rangle_C$ (i.e., which agree in their C entries) and $f(\langle x_i \rangle) = f(\langle x_i' \rangle)$, for every choice of random inputs $\langle r_i \rangle_C$ for C, and every value of the messages $M_C(\langle x_i \rangle, \langle r_i \rangle)$ and $M_C(\langle x_i' \rangle, \langle r_i \rangle)$ received by C:

$$\Pr[\ M_C(\langle x_i \rangle, \langle r_i \rangle) \circ \langle r_i \rangle_C \ \mid \ \langle x_i \rangle\ ] \ = \ \Pr[\ M_C(\langle x_i' \rangle, \langle r_i \rangle) \circ \langle r_i \rangle_C \ \mid \ \langle x_i' \rangle\ ]\ .$$

A protocol F computing f is *private* if any coalition C of size at most $n-1$ gains no additional information from the execution of the protocol. A function f is *private* if there is a private protocol which computes it. It follows that if a private protocol is used to reconstruct the secret, then the reconstruction is secure.

In our interactive secret sharing scheme for the linear functions we use the private protocol of Benaloh [6] to compute the sum over $\mathrm{GF}(q)$ of the shares. For the sake of completeness we describe this protocol:

Private addition over $\mathrm{GF}(q)$**:** $x = \sum_{i=1}^{n} x_i$, $x_i \in \mathrm{GF}(q)$.

Each P_i with input x_i executes:
- chooses $n-1$ random inputs from input $\mathrm{GF}(q)$ denoted $r_{i,1}, \ldots, r_{i,n-1}$.
- computes $r_{i,n} \triangleq x_i - \sum_{j=1}^{n-1} r_{i,j}$.
- sends $r_{i,j}$ to P_j (for every j).

Each P_j computes $y_j = \sum_{i=1}^{n} r_{i,j}$ and sends it to P_1.
Party P_1 computes $\sum_{j=1}^{n} y_j = x$ and sends it to the other parties.