Polynomials Whose Secret Shares Multiplication Preserves Degree for 2-CNF Circuits Over a Dynamic Set of Secrets

Daniel Berend[1,2], Dor Bitan[1], Shlomi Dolev[2]

[1]Department of Mathematics, Ben-Gurion University
[2]Department of Computer Science, Ben-Gurion University

Abstract. One of the most interesting research topics in cryptography is finding efficient homomorphic encryption schemes, preferably information-theoretically secure, which are not based on unproven computational hardness assumptions. The most significant breakthrough in this field was made by Gentry [12] in 2009, and since then, there were various developments.

We suggest here an information-theoretically secure secret sharing scheme that efficiently supports one homomorphic multiplication of secrets in addition to homomorphic additions of, practically, any number of such multiplied secrets. In particular, our scheme enables sharing a dynamic set of secrets amongst N participants, using polynomials of degree $N-1$. Quadratic functions and 2-CNF circuits over the set of secrets can then be homomorphically evaluated, while no information is revealed to any single participant, both before and after the computation. Our scheme is statistically secure against coalitions of less than $N-1$ participants. One possible application of our scheme is a secure homomorphic evaluation of multi-variate quadratic functions and 2-CNF circuits.

1 Introduction

Consider the following scenario. A user is holding some data and wishes to outsource the storage of this data to an untrusted server while enabling the server to perform computations over the data. A vast amount of papers were written on this problem in the past four decades. This work is based on the information-theoretically secure and distributed approach.

The security of cryptographic schemes may be either information-theoretic or computational. In *information-theoretically secure* schemes, the security of the system is derived purely from information theory and depends neither on the computing power of the adversary nor on any computational hardness assumptions. It is possible in such schemes that some information about the plaintext will be revealed to an adversary by the ciphertext, but that leakage of information can be quantified by statistical tools and may be controlled by an appropriate choice of parameters. Information-theoretically secure schemes, in which there is negligible leakage of information, are often called *statistically information-theoretic secure*, or simply *statistically secure*. Information-theoretically secure systems, in which there is absolutely no leakage of information, are *perfectly secure* schemes. These schemes constitute an important class of cryptographic schemes, in which not only will any adversary be unable to decrypt an encrypted message, but it will gain *absolutely no information* about the plaintext from the ciphertext.

The second type of security is *computational security*. It refers to cryptographic schemes that are based on computational hardness assumptions. Specifically, a certain function f is assumed to be easy to compute but hard to invert. This f is used to construct a cryptographic system, such that breaking the system is strongly related to inverting f. The security of these schemes is based on two unproven assumptions: (a) there is no efficient algorithm that inverts f, (b) the adversary does not have enough computing power to break the system in a reasonable time. Therefore, if an information-theoretically secure scheme provides certain performances, it will be preferred over a computationally secure scheme that provides similar performances. Still, computationally secure cryptographic schemes are considered reliable for many uses, often providing acceptable security, and are used in practice.

Secret sharing is a fundamental cryptographic primitive, in which shares of a secret value are distributed to a set of participants in such a way that only authorized sets of participants can reconstruct the secret, while unauthorized sets cannot gain information about it. The set of all authorized sets of participants is *the access structure* of the scheme. An access structure is, of course, closed under containment (i.e., a set containing an authorized set is also authorized) and thus can be defined by its minimal authorized subsets. Since secret sharing was first introduced by Shamir [19] and Blakley [3] (independently) in 1979, it is an active field of research in cryptography.

In Shamir's secret sharing scheme, the secret s is an element of the finite field $\mathbb{F}_p$ and is shared by a *dealer* amongst a set of N participants (where $p > N$) in the following way. Each participant P_i, $1 \leq i \leq N$, is assigned by the dealer with an element α_i of $\mathbb{F}_p^\times$, where the α_i's are distinct. Random elements a_j of $\mathbb{F}_p$, $1 \leq j \leq t-1$, are picked by the dealer. Let f be the polynomial defined by $f(x) = s + \sum_{j=1}^{t-1} a_j x^j$. Each participant P_i gets the value $f(\alpha_i)$. It was proved by Shamir [19] that, in this way, every group of t participants will be able to reconstruct s, but no group of $t-1$ participants gains any information about s. These properties are derived directly from the fact that a polynomial of degree $t-1$ is uniquely determined by its values at t points. Shamir's secret sharing scheme is one of the most influential schemes [6,7,8,9,10,11], as it is perfect information-theoretically secure. It may be used to build schemes that enable secure outsourcing of computations.

As mentioned above, secret sharing has been an active field of research for almost four decades. Some of the research topics are: finding schemes for general access structures, finding verifiable schemes, reducing the size of the shares, constructing secret-sharing-based cryptographic primitives, and finding schemes with homomorphic properties.

Our research focuses on the last topic. Let us describe the main ideas concerning homomorphic secret sharing. Assume $s_1, \ldots, s_d$ are secrets that were shared amongst several participants $P_1, \ldots, P_N$. Each participant, P_j, is holding a share of each of the secrets, denoted: $s_{1j}, \ldots, s_{dj}$. Denote $s_{\text{add}} = \sum_{i=1}^{d} s_i$. If P_j can use the shares it is holding to construct a value $(s_{\text{add}})_j$, such that it is a share of s_{add}, and such that s_{add} can be reconstructed by an authorized set, the scheme is *additively homomorphic*. Similarly, the scheme is *multiplicatively homomorphic* if $s_{1j}, \ldots, s_{dj}$ can be used by P_j to construct a share $(s_{\text{mult}})_j$ of $s_{\text{mult}} := \Pi_{i=1}^{d} s_i$. If a scheme is both additively and multiplicatively homomorphic, it is *fully homomorphic*. If a scheme supports an arbitrary number of homomorphic additions and a bounded number of homomorphic multiplications (or vice versa) it is *somewhat homomorphic*. Such homomorphic properties of cryptographic systems are important attributes. One prominent property of Shamir's scheme is that it is additively homomorphic. This property is based on the fact that the sum of polynomials of

degree $\leq t-1$ is again a polynomial of degree $\leq t-1$. Unfortunately, since the product of two polynomials of degree $\leq t-1$ is in general of a higher degree, Shamir's scheme is not multiplicatively homomorphic. As the degree of the polynomial gets larger, a larger coalition is required in order to extract the secret.

Homomorphic properties are important attributes not only of secret sharing schemes but also of general cryptographic schemes. In particular, assume π is a cryptographic system, m is a message and c its encryption, denoted $\mathrm{Enc}_\pi(m)$. Let f be a function defined over the message space of π. How, if at all, can c be publicly converted into an encryption of $f(m)$? This interesting question is a main subject of research in cryptography. If we assume that the messages and the ciphertexts are elements of a field or a ring (as is often the case), then a more specific form of that question is as follows. Let $m_1, \ldots, m_d$ be messages, and $c_1, \ldots, c_d$ their encryptions. Can $c_1, \ldots, c_d$ be used to publicly generate $c_{\mathrm{add}} = \mathrm{Enc}_\pi\big(\sum_{i=1}^{d} m_i\big)$ or $c_{\mathrm{mult}} = \mathrm{Enc}_\pi\big(\Pi_{i=1}^{d} m_i\big)$? If it is possible to use $c_1, \ldots, c_d$ to publicly generate $c_{\mathrm{add}} = \mathrm{Enc}_\pi\big(\sum_{i=1}^{d} m_i\big)$ (respectively, $c_{\mathrm{mult}} = \mathrm{Enc}_\pi\big(\Pi_{i=1}^{d} m_i\big)$), then π is *additively homomorphic* (respectively, *multiplicatively homomorphic*). If both tasks can be carried out, then π is a *fully homomorphic encryption* (FHE) system. There are several single-operation homomorphic schemes, such as the RSA cryptosystem, which is multiplicatively homomorphic (but cannot support homomorphic additions, and is only computationally secure), and Shamir's secret sharing scheme, which is additively homomorphic (but cannot support homomorphic multiplications).

Midway between single-operation homomorphic encryption systems and FHE systems are *somewhat homomorphic encryption systems*. These systems are additively homomorphic and also support a bounded number of homomorphic multiplications (or multiplicatively homomorphic and support a bounded number of homomorphic additions). To make this concept clear, we modify the above definitions. In the above definition of an additively (respectively, multiplicatively) homomorphic cryptographic system, d could be arbitrarily large. Now we define a cryptographic system to be d_1-*additively homomorphic* (respectively, d_1-*multiplicatively homomorphic*) if it is additively (respectively, multiplicatively) homomorphic for $d \leq d_1$. Thus, additively homomorphic systems are ∞-additively homomorphic, and multiplicatively homomorphic systems are ∞-multiplicatively homomorphic.

Finding FHE schemes is an active field of research in cryptography. The first to propose a computationally secure FHE method was Gentry [12]. Unfortunately, the time complexity of the current implementations is too high to make the scheme practical. Earlier, Rabin and Ben-Or [17] had suggested a verifiable secret sharing scheme, in which a secret is shared amongst a group of participants in such a way that the secret can be extracted by a sufficiently large set of participants, even if some of them are malicious, as long as the majority are honest. The results are information-theoretically secure, with exponentially small error probability. A protocol for multi-party computation is also suggested in [17], but it requires ongoing communication between the participants.

Sander et al. [19] proposed a system to evaluate NC^1 circuits on encrypted values, considering the following case: Alice holds an input x, and Bob is holding a circuit C. We would like Alice to be able to compute $C(x)$, while keeping her input x private, and Bob keeping his circuit C private as well. A main drawback of the system is that the ciphertext length grows exponentially in the depth of the circuit. Their system is based on random self-reducible probabilistic encryption, which may be either computationally or information-theoretically secure. Either way, under the suggested protocol, Alice's input is computationally private, while Bob's circuit remains information-theoretically private.

Cramer et al. [6] suggested a way to construct a multi-party computation protocol based on any linear secret sharing scheme. Some of their results are information-theoretically secure, while others assume the existence of a trapdoor one-way permutation, or the computation hardness of a certain function, based on the Diffie-Hellman encryption system. Either way, their protocols require ongoing communication between the participants, proportional to the depth of the arithmetic circuit. Specifically, when regarding Shamir's secret sharing scheme, multiplication of secrets is possible only if the encrypting polynomials are taken to be of smaller degree to begin with.

In 2005, Boneh et al. [4] proposed a computationally secure public key encryption scheme and showed how it can be used to evaluate 2-DNF circuits over ciphertexts. Their scheme is somewhat homomorphic – additively homomorphic and 2-multiplicatively homomorphic. Since Gentry published his (theoretical and computationally secure) FHE scheme in 2009, there were further developments in this field. Dolev et al. [11] used Shamir's standard scheme for information-theoretically secure multi-party evaluation of RAM programs, where the parties obliviously run a given RAM program over given input. Their solution is based on secure multi-party computation and requires ongoing communication between parties and an external entity called *reducer*, and hence implies high overhead.

Brakerski and Perlman [5] suggested a computationally secure FHE scheme that can be carried out by an unbounded number of participants. Let N be the number of parties whose ciphertexts have been introduced into

the computation so far (inputs from more participants can join the computation later). They used the multi-key approach and obtained a scheme with fully dynamic properties, $O(N)$ ciphertext expansion, and $O(N)$ space complexity for an atomic homomorphic operation. Unfortunately, their results are only theoretical, as their scheme is time-wise impractical.

We discussed above two definitions of fully homomorphic encryption systems — one regarding a general cryptographic system, and one regarding secret sharing schemes. The main difference between the two is that, while in a private/public key encryption systems the ciphertext c is being held by a single entity, in a secret sharing scheme c is distributed amongst several participants. Moreover, we do not regard the set of all shares as an encryption of m, since jointly, they contain all the information needed to extract m. The shares of m can be regarded as an encryption of m only if held by different entities, or by sets that are not in the access structure of the scheme.

Regarding this last distinction, and to conclude, one may characterize cryptographic schemes according to the following criteria: Is $\mathrm{Enc}_\pi(m)$ held by a single entity (or distributed between several)? Is the scheme fully homomorphic (or somewhat homomorphic / single-operation homomorphic / not supporting any homomorphic operations)? Is the scheme information-theoretically secure (or computationally secure)? Is the scheme practical? No system is known that answers 'yes' to all of the above-mentioned questions. Fully homomorphic systems with a single ciphertext holder, such as Gentry's [12] and Brakerski and Perlman's [5], are neither information-theoretically secure nor are they practical. No information-theoretically secure scheme is fully homomorphic, and no practical scheme is fully homomorphic. The scheme suggested by Boneh et al. [4] is a single-entity, somewhat homomorphic, computationally secure and practical scheme.

Our contribution. The main result we obtain in this work is a new textitfunction sieving method. It yields 1-*homomorphic multiplicative pair*s of polynomials, which enables us to adjust Shamir's secret sharing scheme to support one homomorphic multiplication of secrets, shared using polynomials of degree $N-1$ and carried by N participants, keeping it perfectly secure against an attack of a single curious participant and statistically secure against an attack of a coalition of up to $N-2$ curious participants. We use the *statistical difference* function [13, p. 106] to measure the possible leakage of information against such an attack, and show that it is $\approx \frac{2}{p^{N-1-k}}$, where k is the size of the coalition. Of course, one can support homomorphic multiplications of secrets by taking polynomials of a smaller degree to begin with. For example, one can use Shamir's original scheme to share two secrets amongst four participants using linear polynomials, enabling one homomorphic multiplication of secrets, but in this way, the security will be compromised, since any coalition of two participants can easily determine the exact value of the secrets. In our scheme, for example, we can use cubic polynomials to share secrets among four participants in such a way that no coalition of two participants can find the secrets. Our scheme is based on a sophisticated way of choosing the polynomials in a correlated way. Regarding the questions stated above, the scheme we suggest here is a secret sharing scheme, i.e., $\mathrm{Enc}_\pi(m)$ is distributed between several machines. Our scheme is somewhat homomorphic, as we support one (restricted) multiplication and consecutive additions. Our scheme is practical, and most importantly, it is information-theoretically secure, with perfect security against a single curious participant attack, and statistical security against an attack of a coalition of up to $N-2$ curious participants. Moreover, our scheme enables homomorphic evaluation of quadratic functions and 2-CNF circuits over a dynamic set of secret shares. Of course, one can support homomorphic evaluation of quadratic functions and 2-CNF circuits by sharing, along with each pair of secrets, their product, or using Beaver's pre-processing method (as suggested in [2]). But in this way, if new secrets are expected to be joined with the primary ones, then one must keep all the primary secrets in memory, in order to enable the homomorphic computations over the enlarged set of secrets. Our scheme enables additional secrets to be shared over time, while in each stage: a) quadratic functions and 2-CNF circuits over the new set of secrets can be homomorphically and securely evaluated; b) the dealer is not required to store the values of the already-shared secrets in memory, but only the non-free (secret-independent) coefficients of the polynomial that are meant to be used to encrypt the future secrets.

Organization. In Section 2, we introduce the function sieving method and our scheme for secret sharing and multiplication of two secrets amongst N participants using polynomials of degree $N-1$. In Section 3, we prove the correctness of the scheme and discuss its security against an attack of one curious participant and against an attack of a coalition of up to $N-2$ curious participants. In Section 4, we describe how to use our scheme to perform statistically secure homomorphic evaluation of quadratic functions and 2-CNF circuits over a dynamic set of secret shares. Section 5 presents our conclusions.

2 Homomorphic Multiplication of Secret Shares

In this section, we introduce our secret sharing scheme based on Shamir's secret sharing scheme. The scheme will enable us to share two secrets amongst N participants using polynomials of degree $N-1$, perform one homomorphic multiplication of the secrets and consecutive homomorphic additions with further secrets, without increasing the number of participants required to extract the result. We will show that the scheme has perfect security against an attack of a single participant. We also prove that our scheme is statistically secure against coalitions of up to $N-2$ participants. The security of our scheme is not based on any computational hardness assumption.

We begin with a brief overview of our methods and constructions. Assume s_1 and s_2 are two secrets that were shared by Shamir's scheme amongst N participants, P_j, $1 \leq j \leq N$, using two polynomials of degree $N-1$, f_1 and f_2, respectively. For convenience we denote from now on $n = N-1$. Each P_j holds a share of each of the secrets: $\big(\alpha_j, f_1(\alpha_j)\big)$ and $\big(\alpha_j, f_2(\alpha_j)\big)$. As Shamir's scheme is additively homomorphic, the points $\big(\alpha_j, f_1(\alpha_j) + f_2(\alpha_j)\big)$ for $1 \leq j \leq n+1$ are shares of $s_1 + s_2$. Interpolation of these points will yield the unique polynomial of degree $\leq n$ going through them, which is $f_1 + f_2$, whose value at 0 is $s_1 + s_2$. Now, as Shamir's scheme is not multiplicatively homomorphic, the points $\big(\alpha_j, f_1(\alpha_j) \cdot f_2(\alpha_j)\big)$ are in general not shares of $s_1 \cdot s_2$. The polynomial $f_1 \cdot f_2$ is of degree $\leq 2n$. Hence, $2n+1$ points are required to determine it, so that the $n+1$ points we have do not suffice. I.e., no information regarding the secrets may be gained by the $n+1$ points we have. If one insists on interpolating the points $\big(\alpha_j, f_1(\alpha_j) \cdot f_2(\alpha_j)\big)$, that interpolation will yield some polynomial g of degree $\leq n$. It might be the case, though, that $g(0) = s_1 \cdot s_2$. When does it happen? We seek pairs of polynomials to be used with Shamir's scheme that yield $g(0) = s_1 \cdot s_2$. We call this procedure *function sieving*, and as we will show below, it yields 1-*homomorphic multiplicative pair*s of polynomials, which are pairs of polynomials that meet the required condition. We will show that, given the α_j's, these pairs are independent of the secrets and can be determined according to the other coefficients of the polynomials (i.e., all coefficients except for the free terms, which are the secrets).

2.1 Function sieving

Assume that the field $\mathbb{F}_p$, in which the secrets s_1 and s_2 reside, is such that $p \equiv 1 \pmod{n+1}$. In that case, since $\mathbb{F}_p^\times$ is cyclic, it contains a primitive root of unity of order $n+1$. Let α be such a root. For $1 \leq j \leq n+1$ denote $\alpha_j := \alpha^j$, and assign to each participant P_j the value α_j.

Let $a_i, b_i \in \mathbb{F}_p$, $1 \leq i \leq n$, and consider the polynomials

$$f_1(x) = s_1 + \sum_{i=1}^{n} a_i x^i, \quad f_2(x) = s_2 + \sum_{i=1}^{n} b_i x^i,$$

in $\mathbb{F}_p[x]$. Share the secrets s_1, s_2 amongst the participants using f_1, f_2. Namely, distribute to each P_j the values $f_1(\alpha_j), f_2(\alpha_j)$.

Let

$$y_j = f_1(\alpha_j) \cdot f_2(\alpha_j), \quad 1 \leq j \leq n+1.$$

The pairs $(\alpha_j, y_j) \in \mathbb{F}_p^2$ are $n+1$ distinct points through which the polynomial $(f_1 \cdot f_2)(x)$ passes. Since $f := f_1 \cdot f_2$ is of degree $\leq 2n$, it is uniquely determined by $2n+1$ points. Since there are only $n+1$ points (α_j, y_j), interpolation of them will certainly not yield $(f_1 \cdot f_2)(x)$. Nevertheless, let $g(x)$ be the interpolation polynomial for the $n+1$ points, (α_j, y_j).

Obviously, g is of degree $\leq n$. Since f and g agree on the roots of ψ, we have $g(x) \equiv f(x) \pmod{\psi(x)}$, where

$$\psi(x) = \prod_{j=1}^{n+1} (x - \alpha_j).$$

Since the α_j's are all the roots of unity of order $n+1$, we have

$$\psi(x) = x^{n+1} - 1. \tag{2.1.1}$$

Hence, it is easy to compute g. In fact, denote

$$f(x) = s_1 s_2 + \sum_{i=1}^{2n} c_i x^i.$$

We have $x^{n+1} \equiv 1 \pmod{\psi(x)}$, and therefore

$$g(x) \equiv f(x) \equiv s_1 s_2 + c_{n+1} + \sum_{i=1}^{n} (c_i + c_{n+1+i}) x^i \pmod{\psi(x)}.$$

This in turn implies that $g(0) = s_1 s_2 + c_{n+1}$.

Thus, if we take f_1 and f_2 such that $c_{n+1} = 0$, we get $g(0) = f(0)$. Now, $c_{n+1} = \sum_{i=1}^{n} a_i b_{n+1-i}$. Hence, instead of picking the coefficients of f_1 and f_2 uniformly at random, we pick them in such a way that $c_{n+1} = 0$. This is, in essence, the function sieving process. Instead of using Shamir's secret sharing scheme with random polynomials from $\mathbb{F}_p[x]$, we use it with polynomials f_1, f_2, for which $c_{n+1} = 0$, which compels $g(0) = f(0)$. Such a pair (f_1, f_2) is a 1-*homomorphic multiplicative pair* of polynomials.

We define the set of acceptable coefficients for these pairs

$$\mathcal{V}_p := \left\{ (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathbb{F}_p^{2n} \,\middle|\, \sum_{i=1}^{n} a_i b_{n+1-i} = 0, \quad \overline{a} \neq \overline{0} \neq \overline{b} \right\} \cup \left\{ \overline{0} \in \mathbb{F}_p^{2n} \right\},$$

where $\overline{a} = (a_1, \ldots, a_n)$ and $\overline{b} = (b_1, \ldots, b_n)$.[1]

Next, since elements should be picked from $\mathcal{V}_p$, we must define a probability measure on it. First, we compute the cardinality of $\mathcal{V}_p$.

Proposition 1: $|\mathcal{V}_p| = (p^n - 1)(p^{n-1} - 1) + 1$.

Proof: The element $\overline{0} \in \mathbb{F}_p^{2n}$ contributes 1 to $|\mathcal{V}_p|$. The n-tuple $(a_1, \ldots, a_n)$ may be chosen in $p^n - 1$ different ways. For each of these, the n-tuple $(b_1, \ldots, b_n)$ is required to satisfy

$$\sum_{i=1}^{n} a_i b_{n+1-i} = 0.$$

Since $(a_1, \ldots, a_n) \neq \overline{0}$, this equation has $p^{n-1} - 1$ non-zero solutions $(b_1, \ldots, b_n)$. All in all, we get $(p^n - 1)(p^{n-1} - 1) + 1$ elements in $\mathcal{V}_p$. $\square$

Define a probability measure Q on $\mathcal{V}_p$ by:

$$Q(v) = \begin{cases} \frac{1}{p^n}, & v = \overline{0} \in \mathbb{F}_p^{2n}, \\ \frac{1}{p^n(p^{n-1}-1)}, & v \neq \overline{0}. \end{cases}$$

One verifies readily, using Proposition 1, that Q is indeed a probability.

The set $\mathcal{V}_p$ and the probability measure Q are used in the next section, where we present the multiplication scheme.

[1] Each of the $\overline{0}$'s refers to the zero vector of the vector space it belongs to.

2.2 The scheme

We now present our secret sharing scheme. A single homomorphic multiplication of two secrets is supported, to which further secrets can be added homomorphically. Assume a dealer D has two secrets $s_1, s_2 \in \mathbb{F}_p$ and private connection channels with N participants P_i, $1 \leq j \leq N$. As a preliminary phase, the dealer D assigns to each participant P_j an $\alpha_j = \alpha^j \in \mathbb{F}_p^\times$, where α is a primitive root of unity of order N. The scheme stages are as follows:

Algorithm 1: Sharing secrets and computing product

1. The dealer D picks an element $(a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$ according to the distribution Q.
2. D sets $f_1(x) = s_1 + \sum_{i=1}^{n} a_i x^i$ and distributes the value $f_1(\alpha_j)$ to each participant P_j.
3. D sets $f_2(x) = s_2 + \sum_{i=1}^{n} b_i x^i$ and distributes the value $f_2(\alpha_j)$ to each participant P_j.
4. Each P_j computes $y_j = f_1(\alpha_j) \cdot f_2(\alpha_j)$ in $\mathbb{F}_p$ and sends y_j back to D.
5. D finds the unique polynomial $g(x)$ over $\mathbb{F}_p$ that goes through (α_j, y_j).
6. D calculates $s = g(0)$.

As one can see, we use here a polynomial of degree n to represent each of the secrets, and yet we are able to reconstruct their product with only $n+1$ participants (versus $2n+1$ that would be needed originally).

Regarding stage 1 of the protocol, a simple way to Q-pick a suitable element is to create an array with the elements of the set $\mathcal{V}_p$ and insert the element $\overline{0} \in \mathbb{F}_p^{2n}$ into the array $p^{n-1} - 2$ more times. Then, picking an element uniformly at random from that array is equivalent to Q-picking an element of $\mathcal{V}_p$. [2] In stage 5, since $1 \leq i \leq n+1$, the polynomial g is obviously of degree $\leq n$.[3]

We note that one may use Beaver triples, as suggested by Beaver in [2], to support homomorphic multiplication of secret shares, but this method requires sharing, along with the secret, an ancillary correction term to be used by the parties in the multiplication stage. This results in doubling the space complexity of the scheme.

3 The Main Results

The scheme correctness. We now prove the correctness of the scheme. Namely, we prove the following proposition:

Proposition 2: *The value s, calculated at stage 6 of Algorithm 1, is equal to $s_1 \cdot s_2$.*

Proof: The proposition follows directly from the function sieving process, described in Section 2. The coefficients of the polynomials f_1, f_2 were picked from $\mathcal{V}_p$, and hence $\sum_{i=1}^{n} a_i b_{n+1-i} = 0$. By (2.1.1), the α_js were picked in such a way that $\psi(x) = x^{n+1} - 1$. In stage 5 of the scheme, the dealer finds a polynomial g of degree $\leq n$ such that $g(\alpha_j) = y_j$ for $1 \leq j \leq n+1$. This implies that

$$\begin{aligned} g(x) \equiv (f_1 \cdot f_2)(x) &= s_1 s_2 + \sum_{i=1}^{2n} c_i x^i \\ &\equiv s_1 s_2 + c_{n+1} + \sum_{i=1}^{n} (c_i + c_{n+1+i}) x_i \pmod{\psi(x)}. \end{aligned}$$

Hence $g(0) = s_1 s_2 + c_{n+1} \equiv s_1 s_2 \pmod{\psi(x)}$. $\square$

Note that g may now be treated as if it was originally used to share $s_1 \cdot s_2$ amongst N participants, since each of them is now holding y_j. Hence, further secrets can be shared and homomorphically added to $s_1 \cdot s_2$ as in Shamir's standard scheme.

[2] Clearly, one can use the proof of Proposition 1 to implement stage 1 in time $O(n)$.

[3] In fact, given the y_j's, $g(0)$ can be computed without finding g. That procedure is not of our main interests.

The scheme security. We now analyze the scheme security against curious participants' attacks. We will show it has perfect security against one participant attack and statistical security against an attack of a coalition of size up to $N-2$, and compute the statistical difference. To conclude such arguments, first, we must make our assumptions clear. We assume the following:

- **Assumption 1:** The pair of secrets $(s_1, s_2) \in \mathbb{F}_p^2$ is arbitrary. To be precise, we assume they are picked according to an arbitrary distribution Γ, on which we have no assumptions.
- **Assumption 2:** The prime p, the distribution Γ, the set $\mathcal{V}_p$ and the distribution Q over it are public. Namely, if we denote by S_1 and S_2 the $\mathbb{F}_p$-valued random variables indicating the Γ-picked secrets, then the probability $P[(S_1, S_2) = (s_1, s_2)]$ is known for each pair $(s_1, s_2) \in \mathbb{F}_p^2$.
- **Assumption 3:** The element $(a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$, that is Q-picked during stage 1 of the scheme, is kept secret. So are the values $f_1(\alpha_j)$ and $f_2(\alpha_j)$, $1 \leq j \leq N$, that D sends to each participant P_j at stages 2 and 3 of the scheme. In the single participant attack scenario, P_j does not know $f_1(\alpha_i)$ and $f_2(\alpha_i)$ for $i \neq j$. In the scenario of an attack of a coalition of k participants, we assume, without loss of generality, that $P_1, \ldots, P_k$ are curious participants that join their shares in an attempt to find the secrets, but they do not know the shares of other participants.

3.1 Perfect security against single participant attack

In order to show that our scheme has perfect security against one curious participant attack, we need to show that, when P_j receives information from D during stages 2 and 3 of the scheme, he gains absolutely no information about the values of s_1 and s_2. We can summarize the information that P_j receives during stages 2 and 3 of the scheme by the following equations:

$$\begin{aligned} s_1 + \sum_{i=1}^{n} a_i \alpha_j^i &= y_j, \\ s_2 + \sum_{i=1}^{n} b_i \alpha_j^i &= y'_j. \end{aligned} \tag{3.1.1}$$

The unknowns in these equations are s_1, s_2, a_i and b_i, $1 \leq i \leq n$, while all other quantities are known parameters to P_j. We start with

Theorem 1: For an arbitrary fixed $\alpha \in \mathbb{F}_p^\times$ denote $u = \begin{pmatrix} \sum_{i=1}^n a_i \alpha^i \\ \sum_{i=1}^n b_i \alpha^i \end{pmatrix}$. Under the above assumptions, $P[u = \binom{x}{y}] = \frac{1}{p^2}$, for every $\binom{x}{y} \in \mathbb{F}_p^2$.

Proof of Theorem 1: Call u *the result vector*. Since p and α are set, u depends only on the Q-choice of $v \in \mathcal{V}_p$. For $v = (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$, denote:

$$M_v = \begin{pmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \end{pmatrix} \in M_{2\times n}(\mathbb{F}_p).$$

We define a mapping $\mu_\alpha : \mathcal{V}_p \to \mathbb{F}_p^2$ by

$$\mu_\alpha(v) = M_v \begin{pmatrix} \alpha \\ \vdots \\ \alpha^n \end{pmatrix}.$$

For convenience denote $\mu = \mu_\alpha$. Thus,

$$P[u = \tbinom{x}{y}] = P[\mu(v) = \tbinom{x}{y}].$$

To compute $P[u = \binom{x}{y}]$, we first partition $\mathbb{F}_p^2$ into four subsets U_j , $1 \leq j \leq 4$:

- $U_1 = \{\binom{0}{0}\} \subset \mathbb{F}_p^2$.
- $U_2 = \{\binom{x}{0} \in \mathbb{F}_p^2 \mid x \neq 0\}$.

- $U_3 = \{\binom{0}{y} \in \mathbb{F}_p^2 \mid y \neq 0\}$.
- $U_4 = \{\binom{x}{y} \in \mathbb{F}_p^2 \mid x \neq 0, y \neq 0\}$.

We will compute $P[u = \binom{x}{y}]$ for $\binom{x}{y} \in U_j$ for each j separately.

Starting with $j = 1$. We look for elements $v \in \mathcal{V}_p$ such that:

$$\mu(v) = \binom{0}{0}. \tag{3.1.2}$$

Of course, $v = \overline{0} \in \mathbb{F}_p^{2n}$ is a solution of (3.1.2). Assume $v = (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$ is such that $v \neq \overline{0}$ and M_v is a solution of (3.1.2). Namely:

$$\begin{aligned} &\text{I} && \sum_{i=1}^{n} a_i \alpha^i = 0, \\ &\text{II} && \sum_{i=1}^{n} b_i \alpha^i = 0, \\ &\text{III} && \sum_{i=1}^{n} a_i b_{n+1-i} = 0, \end{aligned} \tag{3.1.3}$$

where $(a_1, \ldots, a_n) \neq \overline{0} \neq (b_1, \ldots, b_n)$. Each solution for (3.1.3) gives a suitable element of $\mathcal{V}_p$. Now, (3.1.3)I is a linear equation in n variables a_i. Since the trivial solution is not acceptable, it has $p^{n-1} - 1$ possible solutions $(a_1, \ldots, a_n)$. For each of these solutions, (3.1.3)II-(3.1.3)III is a linear system of two equations in n variables b_i. If the equations are independent, the system has $p^{n-2} - 1$ non-trivial solutions $(b_1 \ldots, b_n)$. Can they be dependent? If they are, there is a $c \in \mathbb{F}_p$ such that $c \cdot \alpha^i = a_{n+1-i}$ for $1 \leq i \leq n$. By (3.1.3)I we get then $\sum_{i=1}^{n} c \cdot \alpha^{n+1-i} \cdot \alpha^i = 0$, so that $n \cdot c \cdot \alpha^{n+1} = 0$. Each of the factors is non-zero, and hence (3.1.3)II-(3.1.3)III are independent. All in all, we get $(p^{n-1} - 1)(p^{n-2} - 1)$ solutions $(a_1, \ldots, a_n, b_1, \ldots, b_n) \neq \overline{0}$.

We conclude that

$$P[u = \binom{0}{0}] = 1 \cdot Q(\overline{0}) + (p^{n-1} - 1)(p^{n-2} - 1) \cdot Q(v_0),$$

where v_0 is any non-zero element of $\mathcal{V}_p$. That is

$$P[u = \binom{0}{0}] = 1 \cdot \frac{1}{p^n} + \frac{(p^{n-1} - 1)(p^{n-2} - 1)}{p^n(p^{n-1} - 1)} = \frac{1}{p^2}.$$

We move to U_2. Thus, we are looking for elements $v \in \mathcal{V}_p$ such that

$$\mu(v) = \binom{x}{0}, \quad (x \neq 0). \tag{3.1.4}$$

Similarly to the computation of $\left|\mu^{-1}\left(\binom{0}{0}\right)\right|$, we get the system

$$\begin{aligned} &\text{I} && \sum_{i=1}^{n} a_i \alpha^i = x, \\ &\text{II} && \sum_{i=1}^{n} b_i \alpha^i = 0, \\ &\text{III} && \sum_{i=1}^{n} a_i b_{n+1-i} = 0, \end{aligned} \tag{3.1.5}$$

where $(a_1, \ldots, a_n) \neq \overline{0} \neq (b_1, \ldots, b_n)$, $x \neq 0$, and each solution of (3.1.5) gives a suitable element of $\mathcal{V}_p$. (3.1.5)I is a non-homogenous linear equation in n variables a_i, and hence has p^{n-1} solutions, $\overline{0}$ is not one of which. For each of these solutions, (3.1.5)II-(3.1.5)III is a system of two linear equations in n variables b_i. If they are independent, it

has $p^{n-2}-1$ non-zero solutions for b_i. Assume they are dependent. Hence, there is $c \in \mathbb{F}_p$ such that $c \cdot \alpha^{n+1-i} = a_i$ for $1 \leq i \leq n$. By (3.1.5)I we get then $\sum_{i=1}^{n} c \cdot \alpha^{n+1-i} \cdot \alpha^i = x$. Then $n \cdot c \cdot \alpha^{n+1} = x$, which gives $c = xn^{-1}$. Hence, there is exactly one solution a_i for (3.1.5)I that yields dependent equations (3.1.5)II-(3.1.5)III. Namely, for $a_i = c \cdot \alpha^{-i} = xn^{-1}\alpha^{-i}$ the system (3.1.5)II-(3.1.5)III is dependent, and hence has $p^{n-1}-1$ non-zero solutions. All in all, we get that

$$\left|\mu^{-1}\left(\binom{x}{0}\right)\right| = (p^{n-1}-1)\cdot(p^{n-2}-1) + 1\cdot(p^{n-1}-1) = p^{n-2}(p^{n-1}-1).$$

We use that and the fact that the trivial solution is not in $\mu^{-1}\left(\binom{x}{0}\right)$ to compute

$$P[u = \tbinom{x}{0}] = P[\mu(v) = \tbinom{x}{0}] = P[v \in \mu^{-1}(\tbinom{x}{0})] = \frac{p^{n-2}(p^{n-1}-1)}{p^n(p^{n-1}-1)} = \frac{1}{p^2}.$$

The computation of $P[u = \binom{x}{y}]$ for $\binom{x}{y} \in U_3$ is analogous, which implies $P[u = \binom{0}{y}] = \frac{1}{p^2}$ for $y \neq 0$.

Now, knowing $\left|\mu^{-1}(U_j)\right|$ for $1 \leq j \leq 3$, we subtract from $\left|\mathcal{V}_p\right|$ and get $\left|\mu^{-1}(U_4))\right| = (p-1)^2 \cdot p^{n-2}(p^{n-1}-1)$. Observe that so far, for a specific $j \in \{1,2,3\}$, all elements of U_j had the same size of preimage under μ. If we show that the same holds for U_4 as well, then together with the fact that $\left|U_4\right| = (p-1)^2$ we get that $\left|\mu^{-1}\left(\binom{x}{y}\right)\right| = p^{n-2}(p^{n-1}-1)$ for $\binom{x}{y} \in U_4$. This in turn will imply that

$$P[u = \tbinom{x}{y}] = P[\mu^{-1}(v) = \tbinom{x}{y}] = p^{n-2}(p^{n-1}-1)\cdot Q(v) = \frac{p^{n-2}(p^{n-1}-1)}{p^n(p^{n-1}-1)} = \frac{1}{p^2}$$

for $\binom{x}{y} \in U_4$. Thus, all that is left is to show is that all elements of U_4 actually have the same size of preimage under μ.

To this end, we define a family of transformations $T_{k,l}$ over $\mathcal{V}_p$. For arbitrary fixed $k, l \in \mathbb{F}_p^{\times}$, let $T_{k,l} : \mathcal{V}_p \to \mathcal{V}_p$ be defined by:

$$T_{k,l}(a_1, \ldots, a_n, b_1, \ldots, b_n) = (ka_1, \ldots, ka_n, lb_1, \ldots, lb_n).$$

The map $T_{k,l}$ is clearly bijective. In fact, the number and positions of zeros in v (if any) are the same as in $T_{k,l}(v)$. The set $\mathcal{V}_p$ and some of its subsets have important properties regarding $T_{k,l}$:

- $\mathcal{V}_p$ **is $T_{k,l}$-invariant** : If $v = (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$, then $\sum_{i=1}^{n} a_i b_{n+1-i} = 0$. It immediately follows that $T_{k,l}(v) = kl\sum_{i=1}^{n} a_i b_{n+1-i} = 0$. Hence $T_{k,l}(v)$ is indeed in $\mathcal{V}_p$.
- **The sets $\mu^{-1}(U_j)$ are $T_{k,l}$-invariant:** If $v = (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mathcal{V}_p$, and $\mu(v) \in U_j$ for a certain j, then:

  $$\mu(v) = \begin{pmatrix} a_1 & \ldots & a_n \\ b_1 & \ldots & b_n \end{pmatrix} \begin{pmatrix} \alpha \\ \vdots \\ \alpha^n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} a_1\alpha^i \\ \sum_{i=1}^{n} b_1\alpha^i \end{pmatrix} \in U_j.$$

  We have:

  $$\mu\big(T_{k,l}(v)\big) = \begin{pmatrix} ka_1 & \ldots & ka_n \\ lb_1 & \ldots & lb_n \end{pmatrix} \begin{pmatrix} \alpha \\ \vdots \\ \alpha^n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} ka_1\alpha^i \\ \sum_{i=1}^{n} lb_1\alpha^i \end{pmatrix}.$$

  Then

  $$\mu\big(T_{k,l}(v)\big) = \begin{pmatrix} k\sum_{i=1}^{n} a_1\alpha^i \\ l\sum_{i=1}^{n} b_1\alpha^i \end{pmatrix}.$$

  Since $k, l \neq 0$, an entry of $\mu(v)$ vanishes if and only if the corresponding entry of $\mu\big(T_{k,l}(v)\big)$ does. Namely, if $\mu(v) \in U_j$, then $\mu\big(T_{k,l}(v)\big) \in U_j$. We conclude that the sets $\mu^{-1}(U_j) \subseteq \mathcal{V}_p$ are invariant under $T_{k,l}$.

Now, let $\binom{x}{y}, \binom{x'}{y'} \in U_j$ for some $1 \leq j \leq 4$. Take $v = (a_1, \ldots, a_n, b_1, \ldots, b_n) \in \mu^{-1}(\binom{x}{y})$. We have $\mu(v) =$

$\begin{pmatrix} \sum_{i=1}^n a_1\alpha^i \\ \sum_{i=1}^n b_1\alpha^i \end{pmatrix} = \binom{x}{y}$. Put:

$$k = \begin{cases} \frac{x'}{x}, & x \neq 0, \\ 1, & x = 0, \end{cases}, \qquad l = \begin{cases} \frac{y'}{y}, & y \neq 0, \\ 1, & y = 0. \end{cases}$$

We get $\mu\big(T_{k,l}(v)\big) = \begin{pmatrix} k\sum_{i=1}^n a_1\alpha^i \\ l\sum_{i=1}^n b_1\alpha^i \end{pmatrix} = \begin{pmatrix} \frac{x'}{x}x \\ \frac{y'}{y}y \end{pmatrix} = \binom{x'}{y'}$. Thus, for every $v \in \mu^{-1}\big(\binom{x}{y}\big)$ we have $T_{k,l}(v) \in \mu^{-1}\big(\binom{x'}{y'}\big)$ for appropriate k, l. This implies that $\big|\mu^{-1}\big(\binom{x}{y}\big)\big| = \big|\mu^{-1}\big(\binom{x'}{y'}\big)\big|$ for $\binom{x}{y}, \binom{x'}{y'} \in U_j$. To conclude, for a given j, all elements of U_j have the same probability. □

We use Theorem 1 to prove the perfect security of our scheme in this scenario. We claim now

Proposition 2: $P[(S_1, S_2) = (s_1, s_2) \mid (3.1.1)] = P[(S_1, S_2) = (s_1, s_2)]$.

Proof: Denote:

$$\theta = P[(S_1, S_2) = (s_1, s_2) \mid (3.1.1)].$$

Explicitly[4],

$$\theta = P\left[(S_1, S_2) = (s_1, s_2) \,\middle|\, \begin{smallmatrix} s_1 + \sum_{i=1}^n a_i\alpha^i = y \\ s_2 + \sum_{i=1}^n b_i\alpha^i = y' \end{smallmatrix}\right]$$

Hence

$$\begin{aligned} \theta &= P\Big[(S_1, S_2) = (s_1, s_2) \Big| u = \begin{pmatrix} y - s_1 \\ y' - s_2 \end{pmatrix}\Big] \\ &= \frac{P\Big[(S_1, S_2) = (s_1, s_2) \cap u = \begin{pmatrix} y - s_1 \\ y' - s_2 \end{pmatrix}\Big]}{P\Big[u = \begin{pmatrix} y - s_1 \\ y' - s_2 \end{pmatrix}\Big]}. \end{aligned}$$

According to Theorem 1, we have $P\big[u = \binom{x}{y}\big] = \frac{1}{p^2}$. Hence, the values of u are independent of (S_1, S_2), so that

$$\theta = \frac{P[(S_1, S_2) = (s_1, s_2)] \cdot \frac{1}{p^2}}{\frac{1}{p^2}} = P[(S_1, S_2) = (s_1, s_2)].$$

□

3.2 Security against coalitions of $k < N - 1$ curious participants

We now turn to analyze the scheme's security against a coalition of k participants for $k < N - 1$. Without loss of generality, we consider the coalition $\{P_1, \ldots, P_k\}$. We will refer to this coalition as *the adversary*. As in the preceding scenario, we can summarize the information the adversary is holding by the system of $2k$ equations:

$$\begin{aligned} s_1 + \sum_{i=1}^n a_i\alpha_1^i = y_{11}, \quad \ldots \quad , s_1 + \sum_{i=1}^n a_i\alpha_k^i = y_{1k}, \\ s_2 + \sum_{i=1}^n b_i\alpha_1^i = y_{21}, \quad \ldots \quad , s_2 + \sum_{i=1}^n b_i\alpha_k^i = y_{2k} \end{aligned} \tag{3.2.1}$$

The unknowns in these equations are a_i, b_i, s_1, s_2, while all other parameters are known to the adversary. We will now prove two useful results concerning this scenario. First, given (3.2.1), all p^2 options for $(s_1, s_2) \in \mathbb{F}_p^2$ are possible. Second, given a pair of secrets, the shares $y_{11}, \ldots, y_{1k}, y_{21}, \ldots, y_{2k}$ distribute almost uniformly. We will soon make this statement precise by analyzing how the matrix $\begin{pmatrix} y_{11} & \cdots & y_{1k} \\ y_{21} & \cdots & y_{2k} \end{pmatrix}$ is distributed over $M_{2\times k}(\mathbb{F}_p)$, given a pair of secrets

[4] We omit the index j and write α, y, y'.

(s_1, s_2), and show that this distribution is statistically close to the uniform distribution. Let (s_1, s_2) be a pair of secrets, and $Y_{(s_1,s_2)}$ be the $M_{2\times k}(\mathbb{F}_p)$-valued random variable indicating the matrix $\left(\begin{smallmatrix} y_{11} & \cdots & y_{1k} \\ y_{21} & \cdots & y_{2k} \end{smallmatrix}\right)$ induced by (s_1, s_2). We will show that the statistical difference [12] between the distributions $Y_{(s_1,s_2)}$ and the uniform distribution over $M_{2\times k}(\mathbb{F}_p)$ is $\approx \frac{1}{p^{n-k}}$. Since statistical difference is a metric, we will conclude by the triangle inequality that the statistical difference between two such distributions, $Y_{(s_1,s_2)}$ and $Y_{(s_1',s_2')}$, is no more than $\approx \frac{2}{p^{n-k}}$.

To this end, we need the following theorem. Denote

$$U = \begin{pmatrix} \sum_{i=1}^n a_i\alpha_1^i, & \dots & , \sum_{i=1}^n a_i\alpha_k^i \\ \sum_{i=1}^n b_i\alpha_1^i, & \dots & , \sum_{i=1}^n b_i\alpha_k^i \end{pmatrix}.$$

We call U *the result matrix.*

Theorem 2: The distribution of the result matrix is given by

$$P\left[U = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right)\right] = \begin{cases} \frac{1}{p^n} + \frac{(p^{n-k}-1)(p^{n-k-1}-1)}{p^n(p^{n-1}-1)}, & \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{smallmatrix}\right), \\ \frac{p^{n-k-1}(p^{n-k}+p-1)}{p^n(p^{n-1}-1)}, & \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right) \in \Omega, \\ \frac{p^{n-k-1}(p^{n-k}-1)}{p^n(p^{n-1}-1)}, & \text{otherwise,} \end{cases}$$

where Ω is a proper subset of $M_{2\times k}(\mathbb{F}_p)$, with cardinality of $(p^k-1)(p^{k-1}-1)$.

Proof of Theorem 2: Since p and $\alpha_1, \dots, \alpha_k$ are set, the result matrix U depends only on the Q-choice of $v \in \mathcal{V}_p$. Using the same notation for M_v as in the proof of Theorem 1, we state the connection between U and v. For $\alpha_1, \dots, \alpha_k \in \mathbb{F}_p^\times$, we define a mapping $\rho : \mathcal{V}_p \to M_{2\times k}(\mathbb{F}_p)$ by

$$\rho(v) = M_v \begin{pmatrix} \alpha_1 & \cdots & \alpha_k \\ \vdots & & \vdots \\ \alpha_1^n & \cdots & \alpha_k^n \end{pmatrix}.$$

Thus:

$$P[U = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right)] = P[\rho(v) = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right)].$$

Let $\left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right) \in M_{2\times k}(\mathbb{F}_p)$. We compute $P[U = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right)]$ by finding the number of elements $v \in V_p$ for which $\rho(v) = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y_1' & \cdots & y_k' \end{smallmatrix}\right)$, and use the probability Q defined above. These elements are exactly the elements $(a_1, \dots, a_n, b_1, \dots, b_n) \in V_p$ with $(a_1, \dots, a_n) \neq \overline{0} \neq (b_1, \dots, b_n)$ that solve the system of equations

$$\begin{aligned} &\mathrm{I}_1 && \sum_{i=1}^n a_i\alpha_1^i = y_1, \\ &\vdots && \vdots \\ &\mathrm{I}_k && \sum_{i=1}^n a_i\alpha_k^i = y_k, \\ &\mathrm{II}_1 && \sum_{i=1}^n b_i\alpha_1^i = y_1', \\ &\vdots && \vdots \\ &\mathrm{II}_k && \sum_{i=1}^n b_i\alpha_k^i = y_k', \\ &\mathrm{III} && \sum_{i=1}^n a_i b_{n+1-i} = 0. \end{aligned} \tag{3.2.2}$$

We solve (3.2.2) and analyze the number of solutions for given $y_1,\dots,y_k,y'_1,\dots,y'_k$. The sub-system (3.2.2)$\mathrm{I}_1$-...-(3.2.2)$\mathrm{I}_k$ consists of k independent equations with n variables $a_1,\dots,a_n$. Its independence follows from the fact that the matrix of the coefficients $(\alpha_j^i)_{i,j}$ is a sub-matrix of Vandermonde matrix with distinct generators $\alpha_1,\dots,\alpha_k$. Hence, (3.2.2)I_1-...-(3.2.2)I_k has p^{n-k} solutions $(a_1,\dots,a_n)$. For each of them, the system (3.2.2)II_1-...-(3.2.2)II_k-(3.2.2)III consists of $k+1$ equations with n variables $b_1,\dots,b_n$. Is this system independent? The equations (3.2.2)II_1-...-(3.2.2)II_k are independent for the same reason that (3.2.2)I_1-...-(3.2.2)I_k are. Hence, we only need to find out whether (3.2.2)III is dependent of (3.2.2)II_1-...-(3.2.2)II_k. This may happen only if there exist $c_1,\dots,c_k\in\mathbb{F}_p$, such that $a_{n+1-i}=\sum_{j=1}^k c_j\cdot\alpha_j^i$ for all $1\le i\le n$. Replacing i for $n+1-i$ and using the fact that $\alpha_j^{n+1}=1$, we get equivalently that $a_i=\sum_{j=1}^k c_j\cdot\alpha_j^{-i}$. Now, a_i must satisfy (3.2.2)I_1-...-(3.2.2)I_k, so we replace each a_i in (3.2.2)I_1-...-(3.2.2)I_k with $\sum_{j=1}^k c_j\cdot\alpha_j^{-i}$ and get

$$\begin{aligned}
&\mathrm{I}_1 \qquad \sum_{i=1}^n \alpha_1^i\cdot\Big(\sum_{j=1}^k c_j\cdot\alpha_j^{-i}\Big)=y_1,\\
&\vdots \qquad\qquad\qquad \vdots\\
&\mathrm{I}_k \qquad \sum_{i=1}^n \alpha_k^i\cdot\Big(\sum_{j=1}^k c_j\cdot\alpha_j^{-i}\Big)=y_k.
\end{aligned}\tag{3.2.3}$$

Given $y_1,\dots,y_k$, this is a system of k equations with k unknowns $c_1,\dots,c_k$. Write (3.2.3) in the form

$$\begin{aligned}
&\mathrm{I}_1 \qquad \sum_{j=1}^k c_j\cdot\sum_{i=1}^n\Big(\frac{\alpha_j}{\alpha_1}\Big)^i=y_1,\\
&\vdots \qquad\qquad\qquad \vdots\\
&\mathrm{I}_k \qquad \sum_{j=1}^k c_j\cdot\sum_{i=1}^n\Big(\frac{\alpha_j}{\alpha_k}\Big)^i=y_k.
\end{aligned}\tag{3.2.4}$$

Now,

$$\sum_{i=1}^n\Big(\frac{\alpha_j}{\alpha_l}\Big)^i=\begin{cases}\sum_{i=1}^n 1=n, & j=l,\\ \frac{\alpha_j}{\alpha_l}\cdot\frac{1-\left(\frac{\alpha_j}{\alpha_l}\right)^n}{1-\frac{\alpha_j}{\alpha_l}}=\frac{\alpha_j\cdot\left(1-\left(\frac{\alpha_j}{\alpha_l}\right)^{-1}\right)}{\alpha_l-\alpha_j}=-1, & j\ne l.\end{cases}$$

Hence, we may write (3.2.4) in the form

$$\begin{pmatrix} n & \dots & -1\\ \vdots & \ddots & \vdots\\ -1 & \dots & n\end{pmatrix}\begin{pmatrix}c_1\\ \vdots\\ c_k\end{pmatrix}=\begin{pmatrix}y_1\\ \vdots\\ y_k\end{pmatrix}.\tag{3.2.5}$$

The matrix A on the left-hand side of (3.2.5) has n's on the main diagonal and -1 elsewhere. Namely, it can be generated by cyclic permutations of its first row (or column). A matrix like that is a *circulant matrix*. We compute its determinant using [13] (or directly) to get $\det(A)=(n-k+1)(n+1)^{k-1}$. Since $k<n<p$, we have $\det(A)\ne 0$, and hence A is invertible. Denote $\overline{c}=(c_1,\dots,c_k)^T$ and $\overline{y}$ the result vector of (3.2.5). We solve (3.2.5) to get the unique solution of this system

$$\overline{c}=A^{-1}\overline{y}.\tag{3.2.6}$$

For given $\overline{y}=(y_1,\dots,y_k)^T$, set $\overline{c}=A^{-1}\overline{y}$. Then $\overline{a}_0=(a_1,\dots,a_n)$ with $a_i=\sum_{j=1}^k c_j\alpha_j^{-i}$ is a solution for (3.2.2)I_1-...-(3.2.2)I_k for which the left-hand side of (3.2.2)III is dependent of the left-hand side of (3.2.2)II_1-...-(3.2.2)II_k. Any other solution $(a_1,\dots,a_n)\ne\overline{a}_0$ of (3.2.2)I_1-...-(3.2.2)I_k yields an independent system (3.2.2)II_1-...-(3.2.2)II_k-(3.2.2)III. For such $\overline{a}_0$, the right-hand side of (3.2.2)III will be dependent of the right-hand side of (3.2.2)II_1-...-(3.2.2)II_k if $\sum_{j=1}^k y'_j\cdot c_j=0$. Denoting $(y'_1,\dots,y'_k)^T=\overline{y'}$, we write that condition equivalently as

$\langle \overline{y'}, \overline{c} \rangle = 0$.

To conclude, given $\binom{y_1 \ldots y_k}{y'_1 \ldots y'_k} \in M_{2\times k}(\mathbb{F}_p)$, set $\overline{c} = A^{-1}\overline{y}$ and $\overline{a}_0 = (a_1, \ldots, a_n)$ with $a_i = \sum_{j=1}^{k} c_j \alpha_j^{-i}$. If $\langle \overline{y'}, A^{-1}\overline{y} \rangle = 0$ then $\overline{a}_0$ is a solution of (3.2.2)I$_1$-...-(3.2.2)I$_k$ for which (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III has p^{n-k} solutions. If $\langle \overline{y'}, A^{-1}\overline{y} \rangle \neq 0$ then $\overline{a}_0$ is a solution of (3.2.2)I$_1$-...-(3.2.2)I$_k$ for which (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III has no solutions.

We can now count the total number of solutions $(a_1, \ldots, a_n, b_1, \ldots, b_n)$ of (3.2.2) in each of the following cases.

- **Case 1.** $\overline{y} = \overline{y'} = \overline{0}$.
  In this case, one solution is the trivial solution, $(a_1, \ldots, a_n, b_1, \ldots, b_n) = \overline{0}$. By (3.2.6) we get here $\overline{c} = \overline{0}$, implying $\overline{a}_0 = \overline{0}$. Now, (3.2.2)I$_1$-...-(3.2.2)I$_k$ has p^{n-k} solutions $(a_1, \ldots, a_n)$. The solution $\overline{a}_0$ yields p^{n-k} solutions $(b_1, \ldots, b_n)$ for (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. Amongst them, only $\overline{b} = \overline{0}$ is acceptable, but we have already counted it. So we are left with $p^{n-k} - 1$ solutions $\overline{a}$ for (3.2.2)I$_1$-...-(3.2.2)I$_k$. Each of these yields p^{n-k-1} solutions $\overline{b}$ for (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. The vector $\overline{b} = \overline{0}$ is always one of them, so we omit it. All in all we get a total of $1 + (p^{n-k} - 1)(p^{n-k-1} - 1)$ valid solutions for (3.2.2).

- **Case 2.** $\overline{y} = \overline{0}, \overline{y'} \neq \overline{0}$.
  By (3.2.6) we get again $\overline{c} = \overline{0}$, implying $\overline{a}_0 = \overline{0}$. Since $\overline{y'} \neq \overline{0}$, $\overline{b} = \overline{0}$ is not a solution of (3.2.2)II$_1$-...-(3.2.2)II$_k$, we obtain no valid solutions for $\overline{a}_0 = \overline{0}$. Each of the other $p^{n-k} - 1$ solutions $\overline{a}$ of (3.2.2)I$_1$-...-(3.2.2)I$_k$ yields p^{n-k-1} solutions $\overline{b}$ of (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III, all of which are valid. All in all we get a total of $p^{n-k-1}(p^{n-k} - 1)$ valid solutions for (3.2.2).

- **Case 3.** $\overline{y'} = \overline{0}, \overline{y} \neq \overline{0}$.
  Analogous to Case 2.

- **Case 4.** $\overline{y} \neq \overline{0} \neq \overline{y'}$ with $\langle \overline{y'}, A^{-1}\overline{y} \rangle \neq 0$.
  In this case there are no solutions with $\overline{a} = \overline{0}$ or $\overline{b} = \overline{0}$. Here, $\overline{a}_0$ is a solution for (3.2.2)I$_1$-...-(3.2.2)I$_k$ which yields no solution of (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. For each of the other $p^{n-k} - 1$ solutions of (3.2.2)I$_1$-...-(3.2.2)I$_k$ there are p^{n-k-1} solutions of (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. Hence, we get a total of $p^{n-k-1}(p^{n-k} - 1)$ valid solutions for (3.2.2).

- **Case 5.** $\overline{y} \neq \overline{0} \neq \overline{y'}$ with $\langle \overline{y'}, A^{-1}\overline{y} \rangle = 0$.
  As in the previous case, there are no solutions with $\overline{a} = \overline{0}$ or $\overline{b} = \overline{0}$. Here, $\overline{a}_0$ is a solution of (3.2.2)I$_1$-...-(3.2.2)I$_k$ which yields p^{n-k} solutions of (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. For each of the other $p^{n-k} - 1$ solutions of (3.2.2)I$_1$-...-(3.2.2)I$_k$ there are p^{n-k-1} solutions for (3.2.2)II$_1$-...-(3.2.2)II$_k$-(3.2.2)III. Hence, we get a total of $p^{n-k-1}(p^{n-k} - 1) + p^{n-k} = p^{n-k-1}(p^{n-k} + p - 1)$ valid solutions for (3.2.2).

Denote

$$\Omega = \left\{ \binom{y_1 \ldots y_k}{y'_1 \ldots y'_k} \in M_{2\times k}(\mathbb{F}_p) \middle| \overline{y} \neq \overline{0} \neq \overline{y'}, \langle \overline{y'}, A^{-1}\overline{y} \rangle = 0 \right\}.$$

To compute $|\Omega|$, observe that $\overline{y}$ can be chosen in $p^k - 1$ different ways. For each of these, the condition $\langle \overline{y'}, A^{-1}\overline{y} \rangle = 0$ is a linear equation with p^{k-1} solutions. We omit the trivial solution and get $|\Omega| = (p^k - 1)(p^{k-1} - 1)$. By the definition of Q, the rest follows. □

An immediate consequence of Theorem 2 is that, given (3.2.1), all p^2 options for $(s_1, s_2) \in \mathbb{F}_p^2$ are indeed possible: If the adversary is holding $\begin{pmatrix} y_{11} & \cdots & y_{1k} \\ y_{21} & \cdots & y_{2k} \end{pmatrix} \in M_{2\times k}(\mathbb{F}_p)$, then, for each of the p^2 possible pairs of secrets $(s_1, s_2) \in \mathbb{F}_p^2$, there is a single suitable $\begin{pmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{pmatrix} \in M_{2\times k}(\mathbb{F}_p)$. This matrix is:

$$\begin{pmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{pmatrix} = \begin{pmatrix} y_{11}-s_1 & \cdots & y_{1k}-s_1 \\ y_{21}-s_2 & \cdots & y_{2k}-s_2 \end{pmatrix}.$$

Since all matrices $\begin{pmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{pmatrix} \in M_{2\times k}(\mathbb{F}_p)$ occur with positive probability, the adversary simply does not have enough information in order to determine the secrets. Now, not all elements $\begin{pmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{pmatrix}$ have the same probability. According to Theorem 2, exactly $(p^k - 1)(p^{k-1} - 1) + 1$ out of the p^{2k} elements of $M_{2\times k}(\mathbb{F}_p)$ have a slightly larger

probability. We use the statistical difference function [12] to measure the leakage of information: If (s_1, s_2) is a pair of secrets, we denote by $Y_{(s_1,s_2)}$ the $M_{2\times k}(\mathbb{F}_p)$-valued random variables indicating the matrix $\left(\begin{smallmatrix} y_{11} & \cdots & y_{1k} \\ y_{21} & \cdots & y_{2k} \end{smallmatrix}\right)$ induced by (s_1, s_2), over the Q-picking of v from V_p. We compute the statistical difference $SD(Y_{(s_1,s_2)}, \mathbb{U})$ between the distribution $Y_{(s_1,s_2)}$ and the uniform distribution over $M_{2\times k}(\mathbb{F}_p)$:

$$\begin{aligned} SD(Y_{(s_1,s_2)}, \mathbb{U}) &= \frac{1}{2} \cdot \sum_{Y \in M_{2\times k}(\mathbb{F}_p)} \left| P\big[Y_{(s_1,s_2)} = Y\big] - P\big[\mathbb{U} = Y\big] \right| \\ &= \frac{1}{2} \cdot \sum_{\left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{smallmatrix}\right)} \left| P\left[\left(\begin{smallmatrix} s_1 + \sum_{i=1}^n a_i \alpha_1^i & \cdots & s_1 + \sum_{i=1}^n a_i \alpha_k^i \\ s_2 + \sum_{i=1}^n b_i \alpha_1^i & \cdots & s_2 + \sum_{i=1}^n b_i \alpha_k^i \end{smallmatrix} \right) = \left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{smallmatrix} \right) \right] - \frac{1}{p^{2k}} \right| \\ &= \frac{1}{2} \cdot \sum_{\left(\begin{smallmatrix} y_1 & \cdots & y_k \\ y'_1 & \cdots & y'_k \end{smallmatrix}\right)} \left| P\left[\left(\begin{smallmatrix} \sum_{i=1}^n a_i \alpha_1^i & \cdots & \sum_{i=1}^n a_i \alpha_k^i \\ \sum_{i=1}^n b_i \alpha_1^i & \cdots & \sum_{i=1}^n b_i \alpha_k^i \end{smallmatrix} \right) = \left(\begin{smallmatrix} y_1 - s_1 & \cdots & y_k - s_1 \\ y'_1 - s_2 & \cdots & y'_k - s_2 \end{smallmatrix} \right) \right] - \frac{1}{p^{2k}} \right|. \end{aligned}$$

Using Theorem 2, a straightforward computation yields

$$SD(Y_{(s_1,s_2)}, \mathbb{U}) = \frac{(p^k - p^{k-1} + 2)(p^k - 1)(p^{k-1} - 1)}{p^{2k}(p^{n-1} - 1)} \approx \frac{p^{3k-1}}{p^{2k} \cdot p^{n-1}} = \frac{1}{p^{n-k}}.$$

Since the statistical difference is a metric, by the triangle inequality we get that

$$SD(Y_{(s_1,s_2)}, Y_{(s'_1,s'_2)}) \approx \frac{2}{p^{n-k}}$$

for any couple of distributions induced by pairs of secrets, $(s_1, s_2), (s'_1, s'_2) \in \mathbb{F}_p^2$.

4 Applications

Our scheme can be used to perform homomorphic evaluation of quadratic functions over variables $s_1, \ldots, s_m$, and arbitrarily long 2-CNF circuits. A quadratic function over the variables $s_1, \ldots, s_m$ is of the form

$$F(s_1, \ldots, s_m) = \sum_{1 \le i,j, \le m} r_{ij} s_i s_j + \sum_{k=1}^{m} t_k s_k + c,$$

with $r_{ij}, t_k, c \in \mathbb{F}_p$. There are $p^{\frac{1}{2}(m^2+3m+2)}$ such functions. We can use our scheme to homomorphically evaluate F. For each of the $\frac{m^2+m}{2}$ pairs of variables s_i, s_j, use our scheme to generate a pair of 1-homomorphic-multiplicative-polynomials f_{ij}, f_{ji}, and distribute s_i, s_j amongst the participants. This pre-processing stage requires $O(m^2)$ communication, but now F can be homomorphically evaluated in a straightforward way. Each participant P_l simply evaluates F over its shares of the secrets and sends the result y_l to the dealer. The dealer in turn calculates the polynomial g going through the points (α_l, y_l) with $g(0) = F(s_1, \ldots, s_m)$.

Communication complexity of the aforementioned scheme may be reduced in the cost of lower security parameters. We now show how one can adjust the suggested scheme and achieve a scheme with $O(m)$ cyphertext instead of $O(m^2)$. Pick an element $\overline{v} = (a_1, \ldots, a_n, b_1, \ldots, b_n)$ from $\mathcal{V}_p$ under the condition that $\sum_{i=1}^n a_i \alpha_l^i \neq 0 \neq \sum_{i=1}^n b_i \alpha_l^i$ for $1 \le l \le N$. Pick $k_1, \ldots, k_m, l_1, \ldots, l_m$ from $\mathbb{F}_p$ uniformly at random and set $f_j(x) = s_j + k_j \sum_{i=1}^n a_i x^i$, and $h_j(x) = s_j + l_j \sum_{i=1}^n b_i x^i$, $1 \le j \le m$. Distribute to participant P_l the $2m$ vector $\big(f_1(\alpha_l), \ldots, f_m(\alpha_l), h_1(\alpha_l), \ldots, h_m(\alpha_l)\big)$. Now, each participant evaluates F over his shares of the secrets. The linear parts of F are computed by each participant using either f_k or h_k. The quadratic parts of F are evaluated by each participant as $f_i(\alpha_l) \cdot h_j(\alpha_l)$. This scheme is perfectly secure against a single participant attack, but is insecure against coalitions of two or more participants.

In various applications, the number of variables is growing over time. In that case, the method described

above can be modified to allow new variables to be joined with the primary ones. Explicitly, assume a dealer has $s_1, \ldots, s_m \in \mathbb{F}_p$, and $s_{m+1}, \ldots, s_{m+k}$ are k more variables whose value may not be determined yet, and are expected to be determined and joined with $s_1, \ldots, s_m$ in the future. We wish to share $s_1, \ldots, s_m$ amongst N participants, in a way that: (a) enables homomorphic evaluation of quadratic functions over the m variables; (b) will enable to share, in the future, the k additional variables amongst the participants; (c) will enable homomorphic evaluation of quadratic functions over the $m+k$ variables. We wish to achieve all that without keeping $s_1, \ldots, s_m$ in memory.

We now demonstrate how these dynamic properties are obtained. For each of the pairs of variables s_i, s_j, $1 \leq i \leq m$, $i \leq j \leq m$, use our scheme to generate a 1-homomorphic-multiplicative-pair of polynomials, f_{ij}, f_{ji}, and distribute s_i, s_j amongst N participants. As in the non-dynamic version, quadratic functions over $s_1, \ldots, s_m$ can now be homomorphically evaluated. For each of the pairs s_i, s_j, $1 \leq i \leq m$, $m+1 \leq j \leq m+k$, use our scheme to generate a 1-homomorphic-multiplicative-pair of polynomials, f_{ij}, f_{ji}. Assuming $s_{m+1}, \ldots, s_{m+k}$ are not known yet, for $m+1 \leq j \leq m+k$ let the free coefficient of f_{ji} be zero, and keep f_{ji} in memory. Distribute s_i to the participants using the first of each pair of 1-homomorphic-multiplicative polynomials, i.e., using f_{ij}. Now, when the values of s_j, $m+1 \leq j \leq m+k$, are determined, add each of them to the corresponding polynomial f_{ji}, $1 \leq i \leq m$, and distribute s_j amongst the participants. In addition to that, for each pair of variables s_i, s_j, $m+1 \leq i \leq m+k$, $i \leq j \leq m+k$, generate a 1-homomorphic-multiplicative-pair of polynomials, f_{ij}, f_{ji}, and distribute s_i, s_j amongst the participants. Now, quadratic functions over the $m+k$ variables, $s_1, \ldots, s_{m+k}$, can be homomorphically evaluated in a straightforward way as in the non-dynamic version described above.

A 2-CNF expression over literals $s_1, \ldots, s_m$ is an expression of the form $(s_{i_1} \vee s_{i_2}) \wedge \cdots \wedge (s_{i_{2t-1}} \vee s_{i_{2t}})$. As we work in $\mathbb{F}_p$, we replace the logic values *True* and *False* with the elements 1 and 0 in $\mathbb{F}_p$, respectively (other elements of $\mathbb{F}_p$ are not logically defined). Logic operations are replaced with $\mathbb{F}_p$ operations as follows. Given literals s_1 and s_2, disjunction is implemented by $s_1 + s_2 - s_1 s_2$ and conjunction is considered as addition in $\mathbb{F}_p$. Negation of s_1 is $1 - s_1$. Then, a 2-CNF expression of length $2t$ is a multi-variable quadratic function, and is assigned *True* if the function is evaluated to $t \in \mathbb{F}_p$, and *False* otherwise. There are 2^{2m^2+m} such expressions that can be homomorphically evaluated using our scheme.

5 Conclusions

We have proposed a scheme to perform a multiplication over secret shares without increasing the number of participants required to extract the product. In our scheme, we have dealt with N participants and used polynomials of degree $N-1$. We have shown how to use our scheme to perform homomorphic and secure evaluation of quadratic functions and 2-CNF circuits over a dynamic set of secret shares with $O(m^2)$ ciphertext.

Of course, one can use Shamir's scheme and enable homomorphic multiplication of secrets by just taking the polynomials to be of lower degree to begin with, but such a solution yields a smaller secret share threshold. E.g., if one runs Shamir's standard secret sharing scheme with four participants, and would like to be able to extract a product of two secrets, he/she would be obligated to work with linear polynomials. In that case, if an adversary manages to discover two of the shares of a certain secret, then the secret is revealed. If one tried to work with quadratic polynomials in the standard scheme, then the product polynomial would be of degree 4, and it requires 5 participants to extract the product. In our scheme, even if an adversary manages to reveal two out of four shares of a certain secret, the secret is information-theoretically kept. We proved that each of the participants holding two correlated secret shares gains absolutely no information about the secrets (even if knowing the α_i that was assigned to him/her by the Dealer). We also proved that a coalition of up to $N-2$ curious participants still cannot reveal the exact value of (s_1, s_2), and that the statistical difference is negligible.

Moreover, one can use Shamir's scheme to enable homomorphic evaluation of quadratic functions and 2-CNF circuits over secret shares by sharing, for each pair of secrets, their product. This solution also results in $O(m^2)$ ciphertext, but in this solution, one must keep the secrets in memory in order to allow new secrets to be joined with the primary ones. In our scheme, the primary secrets are not required to be stored in memory once they were shared.

Our scheme can be used to reduce the communication complexity of cryptographic systems as Shamir secret shared database [2], secure multi-party computation [9], secret shared random access machine [11] and in outsourcing of computations, such as in cloud computing, when statistical security may suffice.

The scheme we suggest here is somewhat surprising. We multiply two shares of degree $N-1$-polynomial-shared secrets and manage to extract the product using no more than the N participants we began with, proving it to be

information-theoretically secure. The innovation is in the function sieving method and in the way we built the set $\mathcal{V}_p$ and defined the probability Q over it. Finally, we believe that our approach and proof techniques may open an opportunity for fruitful research on multi-party computation, as well as other applications.

Acknowledgments. This research was partially supported by the Rita Altura Trust Chair in Computer Sciences; the Lynne and William Frankel Center for Computer Science; grant of the Ministry of Science, Technology and Space, Israel, and the National Science Council (NSC) of Taiwan; the Ministry of Foreign Affairs, Italy; the Ministry of Science, Technology and Space, Infrastructure Research in the Field of Advanced Computing and Cyber Security and the Israel National Cyber Bureau, the Milken Families Foundation Chair in Mathematics.
With pleasure, we thank Amos Beimel and Niv Gilboa for useful inputs.

References

[1] Avni, H., Dolev, S., Gilboa, N. and Li, X. (2016). SSSDB: Database with private information search. In *Algorithmic Aspects of Cloud Computing* (pp. 49-61). Springer International Publishing.

[2] Beaver, D. (1991, August). Efficient multi-party protocols using circuit randomization. In Annual International Cryptology Conference (pp. 420-432). Springer, Berlin, Heidelberg.

[3] Blakley, G. R. (1979). Safeguarding cryptographic keys. *Proc. of the 48th National Computer Conference 1979* (pp. 313-317).

[4] Boneh, D., Goh, E. J. and Nissim, K. (2005, February). Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography Conference* (pp. 325-341). Springer, Berlin-Heidelberg.

[5] Brakerski, Z. and Perlman, R. (2016). Lattice-based fully dynamic multi-key FHE with short ciphertexts. In *Annual Cryptology Conference* (pp. 190-213). Springer, Berlin-Heidelberg

[6] Cramer, R., Damgård, I. and Maurer, U. (2000). General secure multi-party computation from any linear secret-sharing scheme. In *Advances in Cryptology-EUROCRYPT 2000* (pp. 316-334). Springer, Berlin-Heidelberg.

[7] Dawson, E. and Donovan, D. (1994). The breadth of Shamir's secret-sharing scheme. *Computers and Security* 13(1) (pp. 69-78).

[8] Dolev, S., Garay, J., Gilboa, N. and Kolesnikov, V. (2009). Swarming secrets. In *47th Annual Allerton Conference on Communication, Control, and Computing* (pp. 1438-1445).

[9] Dolev, S., Gilboa, N. and Li, X. (2015). Accumulating automata and cascaded equations automata for communicationless information-theoretically secure multi-party computation. In *Proceedings of the 3rd International Workshop on Security in Cloud Computing* (pp. 21-29).

[10] Dolev, S., Lahiani, L. and Yung, M. (2007). Secret swarm unit reactive k-Secret sharing. In *Progress in Cryptology-INDOCRYPT* 2007 (pp. 123-137). Springer, Berlin-Heidelberg.

[11] Dolev, S. and Li, Y. (2016). Secret Shared Random Access Machine. In *Algorithmic Aspects of Cloud Computing* (pp. 19-34). Springer International Publishing.

[12] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Symposium on Theory of Computing* (pp. 169-178).

[13] Goldreich, O. (2009). Foundations of cryptography: volume 2, basic applications. Cambridge University Press.

[14] Gray, R. M. (2006). Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory* 2(3) (pp. 155-239).

[15] Macdonald, I. G. (1998). Symmetric functions and Hall polynomials. Oxford University Press.

[16] Micciancio, D. (2010). A first glimpse of cryptography's Holy Grail. *Communications of the ACM* 53(3) (pp. 95-96).

[17] Rabin, T. and Ben-Or, M. (1989). Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing* (pp. 73-85).

[18] Sander, T., Young, A. and Yung, M. (1999). Non-interactive cryptocomputing for NC 1. In *Foundations of Computer Science, 1999, 40th Annual Symposium.* (pp. 554-566). IEEE

[19] Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11) (pp. 612-613).

[20] Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell Labs Technical Journal*, 28(4) (pp. 656-715).