# Atomic Secure Multi-Party Multiplication with Low Communication

Ronald Cramer[1], Ivan Damgård[2], and Robbert de Haan[3]

[1] CWI, Amsterdam & Mathematical Institute, Leiden University, The Netherlands.
URL: http://www.cwi.nl/~cramer, http://www.math.leidenuniv.nl/~cramer.
Email: cramer@cwi.nl

[2] Comp. Sc. Dept., Aarhus University & BRICS, Denmark. Email: ivan@daimi.au.dk

[3] CWI, Amsterdam, The Netherlands. URL: http://www.cwi.nl/~haan.
Email: R.de.Haan@cwi.nl

**Abstract.** We consider the standard secure multi-party multiplication protocol due to M. Rabin. This protocol is based on Shamir's secret sharing scheme and it can be viewed as a practical variation on one of the central techniques in the foundational results of Ben-Or, Goldwasser, and Wigderson and Chaum, Crépeau, and Damgaard on secure multi-party computation. Rabin's idea is a key ingredient to virtually all practical protocols in threshold cryptography.

Given a passive $t$-adversary in the secure channels model with synchronous communication, for example, secure multiplication of two secret-shared elements from a finite field $K$ based on this idea uses one communication round and has the network exchange $O(n^2)$ field elements, if $t = \Theta(n)$ and $t < n/2$ and if $n$ is the number of players. This is because each of $O(n)$ players must perform Shamir secret sharing as part of the protocol. This paper demonstrates that under a few restrictions much more efficient protocols are possible; even at the level of a single multiplication.

We demonstrate a twist on Rabin's idea that enables one-round secure multiplication *with just* $O(n)$ *bandwidth* in certain settings, thus reducing it from quadratic to linear. The ideas involved can additionally be employed in the evaluation of arithmetic circuits, where under appropriate circumstances similar efficiency gains can be obtained.

## 1 Introduction

Given a passive $t$-adversary in the secure channels model with synchronous communication, secure multiplication of two secret-shared elements from a finite field $K$ based on Rabin's idea uses one communication round and has the network exchange $O(n^2)$ field elements, if $t = \Theta(n)$ and $t < n/2$ and if $n$ is the number of players. This is because each of $O(n)$ players must perform Shamir secret sharing as part of the protocol.

[1] Ronald Cramer's research has been partially supported by NWO VICI.

[3] Robbert de Haan's research has been partially funded by the Dutch BSIK/BRICKS project PDC1.

We demonstrate a twist on Rabin's idea that enables one-round secure multiplication *with just $O(n)$ bandwidth*, thus reducing it from quadratic to linear. However, to obtain this efficiency we need to decrease the maximal corruption tolerance, but still $t = \Theta(n)$, i.e., a number of corruptions is tolerated that is still a constant fraction of $n$. Furthermore, we require the finite field $L$ to have a certain property; it should contain a subfield $K$ over which it has an extension degree linear in $n$.

For this result we emphasize that, unlike in previous approaches (such as [10]), the mentioned costs analysis is not amortized, as we consider "single-shot" (or "atomic") secure multiplication only. The techniques involved can provide considerable efficiency gain in certain secure linear algebra computations, such as securely computing the determinant of a matrix and securely solving a linear system of equations, where the chosen field is typically large in order to ensure a small error probability [5], [6].

A main handle that enables the result mentioned above is a theorem that demonstrates that when certain values can be extracted from the shares in a ramp scheme by means of a linear function, several linear functions on these values can be securely computed at the cost of only a single multiplication and using only a single round of communication. We demonstrate how this theorem, together with a technique due to Franklin and Yung [7], can be used to speed up computation over arithmetic circuits.

After discussing the theorem and the main idea behind our variation, we detail some further handles for trade-offs between communication efficiency and corruption tolerance. We also demonstrate similar reductions in communication complexity for secure computation in the presence of an active adversary.

## 2 Rabin's Secure Multiplication Protocol

We consider Rabin 's idea (as explained in [9]) for secure multiplication. This protocol is a key ingredient to virtually all practical protocols in threshold cryptography.

For the moment we focus on the secure channels model with synchronous communication, in the presence of a passive $t$-adversary where $t$ is maximal such that $t < n/2$ and where $n$ is the number of players in the network. Assuming that the network has $(t, n)$-Shamir-sharings of two secret values $a$ and $b$, the protocol allows the network to securely generate a $(t, n)$-Shamir-sharing of the product $a \cdot b$. The technical idea behind this protocol is a simple and elegant reduction from secure multiplication to secure linear computation.

Concretely, let $K$ be a finite field with $|K| > n$. Let $x_1, \ldots, x_n$ be distinct non-zero elements from $K$. Each player $P_i$ has a share $a_i$ in the secret $a$ and a share $b_i$ in the secret $b$. Let $f$ denote the polynomial of degree at most $t$ such that $f(0) = a$ and such that $f(x_i) = a_i$ for all $i$. Similarly, $g$ is the polynomial defining the secret sharing of $b$.

Now note that the values $(a_1 \cdot b_1, \ldots, a_n \cdot b_n)$ are consistent with the polynomial $f \cdot g$, i.e., $(fg)(x_i) = a_i b_i$ for all $i$. Since $fg$ has degree at most $2t$ and since $2t < n$, these values uniquely determine $f \cdot g$, by Lagrange interpolation. Concretely, there exists a (public) linear map $\phi : K^n \longrightarrow K$ such that $\phi(a_1 b_1, \ldots, a_n b_n) = ab$ always.

This reduces secure multiplication to secure linear computation: it is sufficient to compute $\phi$ securely on the secret inputs $a_i b_i$, where $a_i b_i$ is the input of player $P_i$. These inputs can of course be computed locally. So, first the players perform input sharing, i.e., each player $P_i$ $(t, n)$-Shamir-shares $a_i b_i$ among the network, using a polynomial $h_i$. Then each player $P_j$ simply computes locally $\phi(h_1(x_j), \ldots, h_n(x_j))$ as his share in $ab$. The overall result is clearly a $(t, n)$-Shamir-sharing of $ab$, defined by the polynomial $h = \phi(h_1, \ldots, h_n)$. This protocol takes a singe round of communication, and it involves the exchange of $O(n^2)$ elements from $K$.

## 3 Prior Work: Parallel Secure Computation

Franklin and Yung [7] have shown that interesting advantages can be offered in secure computation by relaxing the corruption tolerance level by just a constant fraction of the number of players. They showed an *amortized* cost reduction in communication complexity. More precisely, they assume that the number of corrupted parties $t$ satisfies $t < cn$ where $c$ is a constant less than the standard maximum that can be tolerated in the given scenario (typically 1/2 or 1/3). The same secure evaluation can now be performed on several different inputs in parallel, while the total communication amounts to that of a single secure evaluation.

Although our goals and techniques substantially differ from [7], we do use some of the ideas. We recall their techniques below. Consider for simplicity the secure channels model with a passive adversary, just as in the description of Rabin's idea, though with the following differences.

Let $\hat{t}$ be a positive integer with $\hat{t} < n/2$, and let $k$ be an integer with $1 \leq k \leq \hat{t}$. Define $t = \hat{t} - k$. The finite field $K$ is chosen such that $|K| > n + k$.

First consider the following variation on Shamir's secret sharing scheme. Let the sets $\{x_1, \ldots, x_n\}$ and $\{e_0, \ldots, e_k\}$ be two disjoint sets of distinct elements from $K$.

- Let $a = (u_0, \ldots, u_k)$ be a vector of secret elements from $K$.
- Choose a random polynomial $f(X) \in K[X]$ of degree at most $\hat{t}$ such that

$$f(e_0) = u_0, \ldots, f(e_k) = u_k.$$

- Define the shares as

$$a_1 = f(x_1), \ldots, a_n = f(x_n).$$

Clearly, $\hat{t} + 1$ shares or more jointly determine $f$ and hence the secret vector $a$. As to privacy, it is a straightforward consequence of Lagrange-interpolation that $t$ or fewer shares jointly give no information on the secret vector. So it is a $(t, \hat{t} + 1)$-ramp scheme, with secrets of length $\hat{t} - t + 1$.

Now, $k+1$ secure multiplications of $(u_0v_0, \ldots, u_kv_k)$ can be performed in a very compact manner. Suppose that vectors $a = (u_0, \ldots, u_k)$ and $b = (v_0, \ldots v_k)$ have been secret-shared. Say that the shares in $a$ are $(a_1, \ldots, a_n)$ (with defining polynomial $f$) and the shares in $b$ are $(b_1, \ldots, b_n)$ (with defining polynomial $g$). The network may now obtain a secret-sharing according to the scheme above (and with the same parameters) of the vector $a * b := (u_0v_0, \ldots, u_kv_k)$ as follows.

First we note that for $j = 0, 1, \ldots, k$, it holds that $(fg)(e_j) = u_jv_j$. For a reason similar to the one used in the description of Rabin's idea, there exists linear maps $\phi_j : K^n \longrightarrow K$ such that $u_jv_j = \phi_j(a_1b_1, \ldots, a_nb_n)$ $(j = 0 \ldots k)$.

Each player $P_i$ now simply secret-shares (according to the scheme above, with the same parameters) the vector $(\phi_0(\epsilon_i)a_ib_i, \ldots, \phi_k(\epsilon_i)a_ib_i)$, where $\epsilon_i \in K^n$ is the $i$-th unit vector. Define the polynomial $h(X) = \sum_{i=1}^{n} h_i(X)$, where $h_i$ is the polynomial used by $P_i$ in the sharing step above $(i = 1 \ldots n)$. This polynomial is consistent with the parameters of the scheme, the secret encoded by it is the vector $a * b$ and each player $P_i$ can locally compute his share as $\sum_{j=1}^{n} h_j(x_i)$. We will demonstrate later that there is a more general way to look at this last resharing step (see Theorem 1).

## 4 Ramp Schemes and Share Conversion

We now present a formal definition of (linear) ramp schemes, which can be seen as a generalization of threshold secret sharing schemes.

**Definition 1.** *Let $M_i$ be a $d_i \times e$ matrix for $i = 1, 2, \ldots, n$. For every set $A \subset \{1, 2, \ldots, n\}$, let $M_A$ be the matrix defined by stacking the matrices $(M_i)_{i \in A}$ on top of each other. The scheme defined as such is called a (linear)* $(t, \hat{t}+1)$-ramp scheme *of embedding degree $k+1$ if the following two properties hold:*

- *For any $A \subset \{1, 2, \ldots, n\}$ with $|A| \geq \hat{t}+1$, there are vectors $r_0, r_1, \ldots, r_k$ such that $r_iM_A = u_i$, where $u_i$ is the $i^{th}$ unit vector.*
- *For any $A \subset \{1, 2, \ldots, n\}$ with $|A| \leq t$ and any vector $v = (v_0, v_1, \ldots, v_k)$ there is a vector $\kappa \in KerM_A$ where the first $k+1$ coordinates of $\kappa$ correspond with the coordinates of $v$.*

Ramp schemes are used for secret sharing as follows. Let $s = (s_0, s_1, \ldots, s_k) \in K^{k+1}$ be a secret vector and choose $b = (b_0, b_1, \ldots, b_{e-1}) \in K^e$ at random under the restriction that $b_i = s_i$ for $i = 0, 1, \ldots, k$. Now define $s_i := M_ib \in K^{d_i}$ as the share for the $i^{\text{th}}$ player. Note that the embedding degree of the ramp scheme defines the dimension of the secret space over $K$.

The first condition for ramp schemes is now equivalent to the statement for the corresponding secret sharing scheme that $\hat{t}+1$ or more players can compute every coordinate of the secret vector via a linear combination of their shares. Furthermore, the second condition is equivalent to the statement that for any subset consisting of at most $t$ players every possible secret vector is equally consistent with their shares. Another key point to note is that ramp schemes allow for a "gray zone" between the unqualified and the qualified number of players, which allows the size of the shares to be smaller than the size of the secret.

There is a way to rewrite the scheme due to Franklin and Yung to the notation used above by applying appropriate operations on the columns of a Vandermonde matrix. However, since the representation using polynomials is rather convenient for both their scheme and our scheme from Section 5, we will stick to a polynomial notation for these schemes in the sequel. Naturally, there is also a straightforward way of rewriting our scheme to the formal notation above, which boils down to the elimination of a number of columns from a Vandermonde matrix.

One of the key ingredients of our results is the following theorem, which allows us to convert shares between different types of linear ramp schemes, while at the same time computing a number of linear functions on secret values in the ramp scheme in parallel.

**Theorem 1.** *Assume that the players hold shares $c_1, .., c_n$ in a linear ramp scheme of the secret vector $(s_1, .., s_m)$ – which means there exist linear maps $\phi_j : K^n \longrightarrow K$ such that $s_j = \phi_j(c_1, \ldots, c_n)$ $(j = 0 \ldots m)$ and that the set of all players is qualified in this scheme. Furthermore, let arbitrary linear functions $F_1, .., F_\ell$, $F_i : K^m \to K$, be given. Then in a single round of communication, the shares in this scheme can be transformed into shares in any other linear ramp scheme with secret space of dimension at least $\ell$ with secret vector*

$$(F_0(s_0, \ldots, s_m), \ldots, F_\ell(s_0, \ldots, s_m)),$$

*Furthermore, privacy is maintained for any subset of players for which privacy holds in both of the ramp schemes involved.*

*Proof.* Assume that the functions $F_j$ are $F_j(x_0, \ldots, x_m) := \sum_{w=0}^{m} \mu_w^{(j)} x_w$ for some $\mu_w^{(j)} \in K$ and define $\beta_i^{(j)} := \sum_{w=0}^{m} \mu_w^{(j)} \phi_w(\epsilon_i) c_i$. Note that

$$s_j = \phi_j(c_1, \ldots, c_n) = \sum_{i=1}^{n} \phi_j(\epsilon_i) c_i,$$

so that

$$\begin{aligned} F_j(s_0, \ldots, s_m) &= \sum_{w=0}^{m} \mu_w^{(j)} s_w = \sum_{w=0}^{m} \mu_w^{(j)} \left( \sum_{i=1}^{n} \phi_w(\epsilon_i) c_i \right) \\ &= \sum_{i=1}^{n} \left( \sum_{w=0}^{m} \mu_w^{(j)} \phi_w(\epsilon_i) c_i \right) = \sum_{i=1}^{n} \beta_i^{(j)}, \end{aligned}$$

and that player $i$ can ramp share the vector $(\beta_i^{(0)}, \ldots, \beta_i^{(\ell)})$ in the target scheme, as the coefficients $\beta_i^{(j)}$ only depend on its share $c_i$ and public information. After all players have reshared their shares in this way and the players locally sum up their new shares, they obtain shares in the target scheme with secret vector $(F_0(s_0, \ldots, s_m), \ldots, F_\ell(s_0, \ldots, s_m))$. The privacy claim is straightforward to verify and the result follows. $\square$

In particular, Theorem 1 demonstrates that we can in a single round of communication securely compute any list of linear functions (up to a certain size) on the ramp shared secret values. Combined with the techniques of Franklin and Yung, this is used in Section 8 to enable more efficient evaluation of certain arithmetic circuits. Theorem 1 is later also used in combination with the ramp scheme from Section 5, where the resulting scheme allows to compute products of values in an extension field of $K$ using only shares and communication consisting of elements in $K$.

## 5 Atomic Secure Multiplication: The Main Idea

In [7], the amortized communication complexity of a secure computation is reduced by performing a linear number of multiplications in parallel. The more general techniques described in this section alternatively allow to reduce the *atomic* communication complexity, i.e., the minimum communication complexity required to perform a *single* secure multiplication. In particular, we demonstrate how a decreased maximum corruption tolerance, while still a constant fraction of $n$, allows one to gain a linear factor in communication complexity for a single multiplication. However, for this we require that the finite field that is used in the computation has some additional structure. These techniques can provide considerable efficiency gain, for instance when used as a building block in secure linear algebra computations over large finite extension fields [5, 6].

The technical idea behind our result can be summarized as follows. We use a dedicated ramp scheme, different from the one in [7]. It is defined using an extension field $L$ over $K$, but each share is just a single element from $K$. The secret is an element in $L$, which is represented as a vector of elements from $K$ by fixing a basis of $L$ over $K$ and interpreting $L$ as a vector space over $K$. This way, the information rate of the scheme improves as the degree of $L$ over $K$ increases, but we pay for this by having to decrease the corruption tolerance appropriately.

This approach is additive in the sense that sums of sharings of two elements from $L$ give a sharing of their sum. The relative difficulty lies in the product. We show a variation on Rabin's idea that allows the network to securely compute, in a single round, the vector-representation over $K$ of the product of two elements from $L$, using just $O(n)$ bandwidth. [4] Our idea depends crucially on the properties of our dedicated ramp scheme.

**Definition 2.** *For each integer $d$ with $0 \leq d \leq 2k$ the polynomial $H_d$ is defined as*

$$H_d(X_0, \dots, X_k, Y_0, \dots, Y_k) = \sum_{0 \leq q,r \leq k \;:\; q+r=d} X_q \cdot Y_r.$$

[4] Our results here should be contrasted with those of [4], which deals with low communication secure computation over very small fields, and uses an entirely different technique.

**Definition 3.** *Let $k$ be a non-negative integer and let $\hat{t}$ be an integer with $2k < \hat{t}$. The linear subspace $V_{k,\hat{t}}(K)$ of the vector space of polynomials of degree at most $\hat{t}$ consists of all polynomials $f(X) \in K[X]$ of the form*

$$f(X) = a(X) + R(X) \cdot X^{2k+1},$$

*where $a(X) \in K[X]$ is a polynomial of (formal) degree $k$ and where $R(X) \in K[X]$ is a polynomial of (formal) degree $\hat{t} - 2k - 1$.*

Note the presence of a "gap" in the polynomials. It ensures that after local multiplication of shares none of the higher-term random coefficients in the corresponding product polynomial interferes with the coefficients that results from the lower-term coefficients (which contain the secret vectors). Furthermore, the degree of the polynomials is chosen large enough to ensure that the higher-term coefficients provide sufficient privacy.

Now assume that $2k < \hat{t}$. Thus, $a(X)$ has degree at most $k$ as a polynomial, but its coefficient vector will be taken of length $k+1$ in all the cases. We will sometimes "identify $a(X)$ with its coefficient vector $a$." Similar for $R(X)$. We have the following trivially verified property.

**Lemma 1.** *If $f(X) = a(X)+R(X)\cdot X^{2k+1}$ and $g(X) = b(X)+R'(X)\cdot X^{2k+1} \in V_{k,\hat{t}}(K)$, then*

$$f(X) \cdot g(X) = H_0(a,b) + H_1(a,b) \cdot X + \ldots + H_{2k}(a,b) \cdot X^{2k} + S(X) \cdot X^{2k+1},$$

*where $a, b$ are taken as the coefficient vectors (of length $k+1$) of the corresponding polynomials and where $S(X)$ is a polynomial of degree at most $2\hat{t} - 2k - 1$.*

Now let $L$ be an extension field of $K$ of degree $k+1$, and let $\theta$ be such that

$$L = K(\theta).$$

The fact that $1, \theta, \ldots, \theta^k$ is a basis for the field $L$ as a $k+1$-dimensional $K$-vector space implies the following lemma. Let $a = u_0 + u_1 \cdot \theta + \ldots + u_k \cdot \theta^k \in L$ and $b = v_0 + v_1 \cdot \theta + \ldots + v_k \cdot \theta^k \in L$, with the $u_i$ and $v_j$ elements from $K$.

**Lemma 2.** *With $K$, $\theta$ and $L$ fixed as above, the following holds. There exist linear maps $\chi_j : K^{2k+1} \longrightarrow K$ $(j = 0 \ldots k)$ such that for all $a, b \in L$*

$$ab = \sum_{j=0}^{k} \chi_j(H_0(a,b), \ldots, H_{2k}(a,b)) \cdot \theta^j,$$

*where $a$ and $b$ are given by their respective coordinate vectors $(u_0, \ldots, u_k)$ and $(v_0, \ldots, v_k)$.*

This lemma is easily verified by multiplying everything out, rewriting the powers $\theta^j$ with $j > k$ with respect to the basis chosen and making the substitutions.

Now consider the following secret sharing scheme. It is assumed that $\theta$ is fixed (and public), as well as the other parameters introduced above. A secret can be any element $a \in L$, *represented as a $k+1$-vector of elements from $K$*: $a = u_0 + u_1\theta + \ldots + u_k\theta^k$, with the $u_j$ in $K$. Each share will be an element of $K$ however. Define

$$t = \hat{t} - 2k.$$

1. Let
$$a = u_0 + u_1 \cdot \theta + \ldots + u_k \cdot \theta^k \in L$$
be the secret value.
2. Choose $f(X) \in V_{k,\hat{t}}(K)$ at random such that
$$f(X) = a(X) + R(X) \cdot X^{2k+1},$$
where $a(X) \in K[X]$ is the polynomial of degree at most $k$ whose coefficient vector is $(u_0, \ldots, u_k)$ and where $R(X) \in K[X]$ is a polynomial of degree at most $\hat{t} - 2k - 1$.
3. Set
$$a_1 = f(x_1) \in K, \ldots, a_n = f(x_n) \in K$$
as the shares.
4. For any set $A \subset \{1, \ldots, n\}$ with $|A| \geq \hat{t} + 1$, the reconstruction of $a \in L$ from the shares $\{a_i\}_{i \in A}$ is by standard Lagrange Interpolation.

As for privacy, we note the following. If $|A| \leq t$ $(= \hat{t} - 2k)$, then the collection of shares $\{a_i\}_{i \in A}$ gives no information on the secret $a$. Indeed, for each such set $A$ and for each $z \in L$ there exists a $\kappa(X) \in V_{k,\hat{t}}(K)$ such that

$$\kappa(X) = z(X) + T(X) \cdot X^{2k+1},$$

where $T(X)$ is a polynomial of degree at most $\hat{t} - 2k - 1$, and such that

$$\kappa(x_i) = 0 \text{ for all } i \in A,$$

and this implies the privacy claim, for instance by a simple argument similar to the one used in the analysis of general linear secret sharing. The existence of $\kappa(X)$ per se follows from the lemma below, an immediate consequence of Lagrange's Interpolation Theorem.

**Lemma 3.** *Let $x_1, x_2, \ldots, x_e$ be distinct non-zero elements of $K$. Let $d$ be an integer with $d \geq e$. For any $z_0, \ldots, z_{d-e} \in K$ and for any $y_1, \ldots, y_e$ there exists a polynomial $\kappa(X) \in K[X]$ of degree at most $d$ such that*

$$\kappa(X) = z_0 + z_1 \cdot X + \ldots + z_{d-e} \cdot X^{d-e} + \textit{ higher order terms },$$

*and*

$$\kappa(x_1) = y_1, \ldots, \kappa(x_e) = y_e.$$

*Proof.* Define $f_1(X) = \sum_{j=0}^{d-e} z_j X^j$ and let $f_2(X)$ be the polynomial of degree at most $e-1$ through the $e$ points $(y_i - f_1(x_i))/x_i^{d-e+1}$. Then the polynomial $\kappa(X) = f_1(X) + f_2(X) \cdot X^{d-e+1}$ is the unique polynomial that has the required properties. $\square$

Thus, the dedicated scheme above is a $(t, \hat{t}+1)$ ramp scheme with shares in $K$ and the secret in $L$ (as a vector of length $\frac{\hat{t}-t}{2}+1$ over $K$).

In order to state the claimed secure multiplication protocol we need the following lemma, which can easily be verified using arguments similar to the ones used in standard proofs of Langrange's Interpolation Theorem, or by using the properties of Vandermonde determinants.

**Lemma 4.** *Let $x_1, \ldots, x_{\ell+1}$ be fixed distinct elements of $K$. Then there exist linear maps $\phi_j : K^{\ell+1} \longrightarrow K$ ($j = 0 \ldots \ell$) such that the following holds. Let $y_1, \ldots, y_{\ell+1}$ be any elements of $K$. Let $f \in K[X]$ be the unique polynomial of degree at most $\ell$ such that $f(x_1) = y_1, \ldots, f(x_{\ell+1}) = y_{\ell+1}$. Then*

$$f(X) = \phi_0(y_1, \ldots, y_{\ell+1}) + \phi_1(y_1, \ldots, y_{\ell+1}) \cdot X + \ldots + \phi_\ell(y_1, \ldots, y_{\ell+1}) \cdot X^\ell.$$

Still in the secure channels model as before, assume that $\hat{t} < n/2$. Suppose that values $a = u_0 + u_1 \cdot \theta + \ldots + u_k \cdot \theta^k \in L$ and $b = v_0 + v_1 \cdot \theta + \ldots + v_k \cdot \theta^k \in L$, with coefficients in $K$, have been secret-shared according to the dedicated scheme explained above. Write $f \in K[X]$ for the polynomial defining the sharing of $a \in L$, with respective shares $a_1, \ldots, a_n \in K$, and write $g$ defining that of $b \in L$, with respective shares $b_1, \ldots, b_n \in K$.

It now follows immediately from the fact that $\hat{t} < n/2$ and from Lemmas 1, 2, and 4 that there exist linear maps $\psi_j : K^n \longrightarrow K$ such that

$$ab = \sum_{j=0}^{k} \psi_j(a_1 b_1, \ldots, a_n b_n) \cdot \theta^j \in L.$$

The coefficients defining these linear maps can be computed efficiently. We can now use Theorem 1 to convert the local products of the shares of the players into a sharing of $ab$.

If the degree $[L : K] = k+1$ of the extension field $L$ satisfies the conditions detailed below, we can now achieve $O(n)$ communication.

We have

$$t + 2k = \hat{t} \text{ and } \hat{t} < n/2.$$

So if we set, say,

$$2\hat{t} + 1 = n,$$

and

$$k = cn,$$

for some real constant $c$, then we can achieve $t$ maximal such that

$$t < \frac{(1-\delta)n}{2}, \text{ where } \delta = 4c.$$

If the parameters are such, secure multiplication of two elements from the field $L$ is done with communication $O(n^2)$ elements from $K$, which is equivalent to $O(n)$ elements from $L$. This is as claimed.

### 5.1 A More General View

It is possible to look at the secure multiplication protocols in a more general way, that contain both our results and those of Franklin and Yung [7] as special cases.

Both in the protocol of Franklin and Yung and our protocol from Section 5, the protocols start out with two sets of shares, defining secret vectors $(s_0, ..., s_m), (s'_0, ..., s'_m)$ respectively. We then compute locally the pairwise products of shares in the two vectors and these pairwise products can be seen as shares in a new ramp scheme, different from the original one.

For instance, in the scheme by Franklin and Yung the secret vector defined by the "local products" of the shares is $(s_0 s'_0, s_1 s'_1, \dots, s_m s'_m)$ according to the new scheme defined On the other hand, in the protocol from Section 5, assuming the same initial secret vectors, we obtain secret vector $(\sum_{i+j=0} s_i \cdot s'_j, \sum_{i+j=1} s_i \cdot s'_j, \dots, \sum_{i+j=2m} s_i \cdot s'_j)$ consisting of all homogeneous sums of the secret coefficients. This is why we can obtain different results after the application of Theorem 1.

In general, we can start from any ramp scheme $\mathcal{R}$, do the local multiplications and obtain a sharing of some quadratic function of the two original secret vectors in a new ramp scheme $\mathcal{R}'$ that depends on $\mathcal{R}$. This is not always useful – for instance, it is not always the case that the set of all players is qualified in $\mathcal{R}'$. Franklin/Yung and our scheme are two nicely structured examples, where useful results are indeed obtained.

We note that one can also obtain the homogeneous sums we use by multiple applications of Franklin and Yung's scheme, but since this would require $O(n)$ applications of their scheme (in order to obtain the required cross-products) this would be much less efficient.

## 6 Further Trade-Offs

In Section 5, we presented a scheme which is secure against a $t$-adversary. We now show a variation that is secure against a (stronger) $t'$-adversary with $t' > t$, where $t' - t$ is a constant fraction of $n$. Given again a finite field $L$ with extension degree $k + 1$ over a subfield $K$, the bandwidth requirement remains $O(n)$, but there is a larger hidden constant.

The idea is to introduce a slightly modified version of the dedicated ramp scheme from Section 5. Basically, the coefficients of a secret element $a \in L$ are distributed over two polynomials $f_1$ and $f_2$ with smaller gaps than the polynomial that was used before, and the secure multiplication is then performed with these two polynomials by exploiting cross-products. This doubles the size of the shares and the required bandwidth. There is also a natural generalization of this idea involving more then two polynomials and cross-products of shares.

In Section 5, a $t$-adversary was defined where $t = \hat{t} - 2k$ for some integer $\hat{t} < n/2$ and where $k+1$ is the degree of $L$ over $K$. In this section, we fix the value $\hat{k} = \lceil (k-1)/2 \rceil$, and define the $t'$-adversary by $t' = \hat{t} - 2\hat{k}$. We now explain the details of the variation.

For an arbitrary value $a = u_0 + u_1 \cdot \theta + \ldots + u_k \cdot \theta^k \in L$, with coefficients in $K$, we denote $a^{(1)} = u_0 + u_1 \cdot \theta + \ldots + u_{\hat{k}} \cdot \theta^{\hat{k}}$ and $a^{(2)} = a - a^{(1)}$. Furthermore, we define $a^{(1)}(X) = u_0 + u_1 \cdot X + \ldots + u_{\hat{k}} \cdot X^{\hat{k}}$ and $a^{(2)}(X) = u_{\hat{k}+1} + u_{\hat{k}+2} \cdot X + \ldots + u_k \cdot X^{k-\hat{k}-1}$.

For $i \in \{1, 2\}$, choose $f_i(X) \in V_{\hat{k},\hat{t}}(K)$ at random such that

$$f_i(X) = a^{(i)}(X) + R_i(X) \cdot X^{2\hat{k}+1},$$

where $a^{(1)}(X)$ is the polynomial of formal degree $\hat{k}$ with the initial coefficients $(u_0, u_1, \ldots, u_{\hat{k}})$, $a^{(2)}(X)$ is the polynomial of formal degree $k - \hat{k} - 1$ with the remaining coefficients $(u_{\hat{k}+1}, u_{\hat{k}+2}, \ldots, u_k)$ and where $R_1(X), R_2(X) \in K[X]$ are polynomials of formal degree $\hat{t} - 2\hat{k} - 1$. Then $f_1$ and $f_2$ both encode exactly half of the coefficients of $a$ (if $k$ is odd) or $f_1$ encodes one more coefficient of $a$ than $f_2$ (if $k$ is even). These polynomials are used in this section to perform the secure multiplication.

Assume that a value $b = v_0 + v_1 \cdot \theta + \ldots + v_k \cdot \theta^k \in L$ has likewise been encoded, resulting in polynomials $b^{(1)}(X)$, $b^{(2)}(X)$, $g_1(X)$ and $g_2(X)$, and that every player $P_i$ received the values $a_i^{(1)} = f_1(x_i)$, $a_i^{(2)} = f_2(x_i)$, $b_i^{(1)} = g_1(x_i)$ and $b_i^{(2)} = g_2(x_i)$. By Lemma 3, no subset of $t - 2\hat{k}$ players can obtain any information about $a^{(1)}$, $a^{(2)}$, $b^{(1)}$ or $b^{(2)}$, and therefore the players in such a subset also cannot obtain any information about $a$ or $b$.

We now make use of the observation that

$$(ab)(X) = (a^{(1)}b^{(1)})(X) + (a^{(1)}b^{(2)} + a^{(2)}b^{(1)})(X) \cdot X^{\hat{k}+1} + (a^{(2)}b^{(2)})(X) \cdot X^{2\hat{k}+2},$$

with as coefficients the values $H_0(a, b), H_1(a, b), \ldots, H_{2k}(a, b)$. This is straightforward to verify using the discussion from the last section. Since by Lemma 4 there exists a linear map $\phi_\ell$ such that for $i, j \in \{1, 2\}$ the $\ell^{\text{th}}$ coefficient of $(f_i g_j)(X)$ can be computed as $\phi_\ell(a_1^{(i)} b_1^{(j)}, a_2^{(i)} b_2^{(j)}, \ldots, a_n^{(i)} b_n^{(j)})$, the same holds for $a^{(i)}b^{(j)}(X)$. In particular there exist linear maps $\psi_\ell : K^n \longrightarrow K$ such that

$$ab = \sum_{\ell=0}^{k} \psi_\ell(C_{11}, C_{12}, C_{21}, C_{22}) \cdot \theta^\ell \in L,$$

where $C_{ij} = (a_1^{(i)} b_1^{(j)}, \ldots, a_n^{(i)} b_n^{(j)})$ for $i, j \in \{1, 2\}$. Therefore, the techniques from the previous section can be used to construct a multiplication protocol that leads to two polynomials $h_1(X)$ and $h_2(X)$ of the proper form that encode the coefficients of $ab \in L$.

## 7 Secure MPC Against an Active Adversary (Overview)

Using the new techniques, we construct a protocol for secure multiplication in the presence of an active $t$-adversary that requires only $O(n^2)$ bandwidth when the multiplication is performed in a field $L$ with extension degree $k+1$ over a subfield $K$. Again, the corruption tolerance is not maximal, as we require $t = \hat{t} - 3k$ with $t < \hat{t} < n/4$, but it is still a constant fraction of $n$. Below we sketch the underlying ideas of the protocol. A more detailed description can be found in the appendix.

The obvious weakness of the protocol described in Section 5 is that the outcome completely depends on the polynomials $h_i$ that the players select. Even if only one of these polynomials is not selected according to the protocol specification, the final outcome can encode any arbitrary element of $L$ or not even be of the correct form. However, a closer inspection of the protocol reveals that the values of the leading coefficients of every polynomial $h_i$ mainly depend on the corresponding value $a_i b_i$. Therefore, we can use VSS to let the players secret share their value $a_i b_i$, and then let the players locally compute their shares in polynomials $h_i$ that are guaranteed to be of the proper form. We now sketch the key ingredients of the protocol.

**Dedicated VSS** We use an adaptation of the four-round VSS protocol by Gennaro et al. [8] that allows the players to verify the presence of a gap in a secret sharing polynomial. In particular, we show that it is sufficient if the polynomials that the honest players receive as their shares using this scheme contain the desired gap.

**Resharing step** Every player $P_i$ reshares the value $a_i b_i$ using an instance of the dedicated VSS scheme by embedding it in a secret sharing polynomial $v_i$ of formal degree $\hat{t} - k$ that has a gap of size $2k$ following the constant coefficient. Furthermore, player $P_i$ uses VSS to distribute evaluations on a random polynomial of formal degree $2\hat{t}$ that has a zero constant coefficient. The value $a_i b_i$ is the constant coefficient of a polynomial of formal degree $2\hat{t}$ in which all the players have a share due to the VSS scheme. Therefore, the players can jointly subtract the polynomial $v_i$ from this polynomial and mask the result by adding the random polynomial. These operations can all be performed locally on the shares and lead to shares in the resulting polynomial. The players then publicly reconstruct this polynomial by pooling their shares and verify whether it has a zero constant coefficient. This ensures that player $P_i$ indeed reshared the value $a_i b_i$.

**Local computation** Since the polynomial $h_i(X)$ should contain the element

$$\sum_{j=0}^{k} \psi_j(\epsilon_i) a_i b_i \theta^j \in L,$$

the polynomial

$$h_i(X) = \sum_{j=0}^{k} \psi_j(\epsilon_i) X^j v_i(X)$$

is of the correct form and every player $P_m$ can locally compute a share $h_i(x_m)$ in this polynomial using the share $v_i(x_m)$. The sum of these shares then gives a share in a polynomial of the proper form that encodes the product $ab$.

## 8 Efficient Circuit Evaluation

This section shows another application of Theorem 1. Consider any arithmetic circuit $C$ and a set of inputs to $C$ and suppose that we evaluate the circuit by repeating the following two steps until all the gates have been evaluated:

1. Evaluate all linear gates for which we have both inputs, i.e., the addition gates and gates that perform multiplication by a constant.
2. Evaluate all multiplication gates for which we have both inputs.

Now let $S(C)$ be the minimum number of multiplication gates that are handled in one instance of step 2. We will refer to this value $S(C)$ as the *multiplicative speedup* of $C$.[5] Arithmetic circuits with large multiplicative speedup occur frequently in settings related to secure linear algebra [5]. For instance, constant-round protocols for secure unbounded fan-in multiplication and secure matrix multiplication require many parallel secure multiplications in a single step.

It is a natural idea to apply the scheme of Franklin and Yung here to perform these multiplications in parallel, but in order to do this it is required that the values that are to be multiplied are "aligned" in the corresponding instances of the ramp scheme.

Theorem 1 enables us to perform this aligning and more. If the inputs to the multiplication are available as secrets of some ramp scheme, or even merely available via a linear function on the shares that the players hold in a number of (potentially different) ramp schemes, a single resharing round can be used in order to correctly align the inputs to the parallel multiplications. This also implies that the same resharing step can simultaneously perform the operations required in step 1 before the multiplications are performed, and after local multiplication of the new shares we can continue with the preparations for the next multiplication round. We formulate this consequence of Theorem 1 a bit more precisely below.

**Theorem 2.** *Consider an arithmetic circuit $C$ over the field $\mathbb{F}_q$ with multiplicative speedup $m$. Then there exists a passively secure protocol for $n$ players that securely evaluates $C$ having communication complexity $O(|C|n^2k/m+C')$, where $C'$ is the complexity of sharing the inputs and $k = \log(q)$. The protocol is secure against at most $n/2 - m$ passive corruptions.*

*Proof.* (Sketch) Assume for simplicity that each multiplication layer in $C$ consists of exactly $m$ gates. Then to perform one set of multiplications, the protocol of Franklin and Yung requires ramp sharings, say in ramp scheme $\mathcal{R}$ of two blocks $A$

[5] This term is inspired by [11], where the speedup is defined to be the factor you save in runtime due to parallelism.

and $B$ of $m$ values each, where $A$ contains all the left inputs to the multiplication gates and $B$ contains all the right inputs in matching order. Local multiplication of the shares of $A$ and $B$ then produces a linear secret sharing (in a new scheme $\mathcal{R}'$) of all the outputs from the multiplication gates.

Now note that we can assume that as input to an instance of Step 1 above, we have a linear sharing of all values going into Step 1. This is either obtained because the inputs are shared initially, or we have a sharing in $\mathcal{R}'$ which was output from a previous instance of Step 2. All we need is that the set of all players is qualified in the scheme that occurs here. We now need to subject these values to a linear function and place the results in the blocks $A$ and $B$. Using Theorem 1, we can do exactly this in one round and communication complexity $n^2k$. Clearly, there can be no more than $|C|/m$ multiplication layers, and the scheme of Franklin and Yung that we start from is private as long as there are at most $n/2 - m$ corruptions □

## References


1. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of STOC 1988*, pages 1–10. ACM Press, 1988.
2. G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings Proceedings of National Computer Conference '79*, volume 48 of *AFIPS Proceedings*, pages 313–317, 1979.
3. D. Chaum, C. Crépeau, and I. Damgaard. Multi-party unconditionally secure protocols. In *Proceedings of STOC 1988*, pages 11–19. ACM Press, 1988.
4. H. Chen and R. Cramer. Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computation over Small Fields. In *Proceedings of 26th Annual IACR CRYPTO*, volume 4117, pages 516–531, Santa Barbara, Ca., USA, August 2006. Springer Verlag LNCS.
5. R. Cramer and I. Damgaard. Secure Distributed Linear Algebra in Constant Number of Rounds. In *Proceedings of CRYPTO 2001*, volume 2139, pages 119–136. Springer LNCS, 2001.
6. R. Cramer, E. Kiltz, and C. Padró. A Note on Secure Computation of the Moore-Penrose and Its Application to Secure Linear Algebra. Manuscript, 2006.
7. M. Franklin and M. Yung. Communication complexity of secure computation. In *Proceedings of STOC 1992*, pages 699–710. ACM Press, 1992.
8. R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The Round Complexity of Verifiable Secret Sharing and Secure Multicast. In *Proceedings of STOC 2001*, pages 580–589. ACM Press, 2001.
9. R. Gennaro, M. O. Rabin, and T. Rabin. Simplified VSS and fasttrack multi-party computations with applications to threshold cryptography. In *Proceedings of PODC 1997*, pages 101–111, 1998.
10. M. Hirt and U. Maurer. Robustness for Free in Unconditional Multi-Party Computation. In *Proceedings of CRYPTO 2001*, volume 2139, pages 101–118. Springer LNCS, 2001.
11. C. P. Kruskal, L. Rudolph, and M. Snir. A complexity theory of efficient parallel algorithms. *Theoretical Computer Science*, 71(1):95–132, 1990.
12. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

# A Secure MPC Against an Active Adversary

In this section we describe in detail the key ingredients of the protocol secure against an active adversary, as described in Section 7. Throughout this section, we assume that $t = \hat{t} - 3k$ with $\hat{t} < n/4$.

## A.1 VSS

We start by describing the adaptation of the four-round VSS protocol by Gennaro at al. [8] that allows the players to verify the presence of a gap in the secret sharing polynomial.

Let $f$ be the polynomial defining the sharing of $a \in L$, as described in Section 5. The dealer $D$ randomly selects a symmetric bivariate polynomial $F(X,Y) = \sum_{i,j=0}^{\hat{t}} e_{ij} X^i Y^j \in K[X,Y]$ under the restriction that $F(X,0) = f(X)$ and that $e_{ij} = e_{ji} = 0$ for $j = k+1, k+2, \ldots, 2k$ and $i = 0, 1, \ldots, n$. The dealer $D$ and the players now execute the following steps:

1. $D$ privately sends to every player $P_i$ the polynomial $f_i(X) := F(X, x_i)$ by transmitting the $\hat{t} - k$ coefficients that are not equal to zero by default. In every subset of two players $\{i,j\}$ one of the players (which one can be fixed before execution of the protocol) selects a random pad $r_{ij} = r_{ji}$ and transmits this value privately to the other player in this set.
2. Player $P_i$ broadcasts for every player $P_j$ the value $a_{ij} = f_i(x_j) + r_{ij}$.
3. For every pair $a_{ij} \neq a_{ji}$, the dealer, $P_i$ and $P_j$ each broadcast the value $f_i(x_j) = f_j(x_i) = F(x_i, x_j)$.
   A player is called *unhappy* if his value does not match the dealer's value. If there are more than $t$ unhappy players, the dealer is disqualified and the protocol stops.
4. For every unhappy player $P_i$ the dealer broadcasts $f_i(X)$ and every player $P_j$ that is not unhappy broadcasts the value $f_j(x_i)$.
5. Every player checks for every broadcast polynomial $f_i(X)$ whether at least $3\hat{t}+1$ happy players $P_j$ broadcast a value $f_j(x_i)$ such that $f_i(x_j) = f_j(x_i)$. If this is not the case, the dealer is disqualified. The broadcast polynomials are from here on (publicly) used as the shares of the corresponding players.

As in [8], this protocol has the properties that when the dealer is honest, no new information is disclosed to the adversary after the first round and that when the protocol completes all honest players have obtained consistent polynomials $f_i(X)$. Therefore, the main properties to be verified here are that in the case of an honest dealer no information is disclosed about $a$ in the first round and that in the case of a dishonest dealer the polynomial $f(X)$ that is fixed by the resulting polynomials $f_i(X)$ is of the proper form. Note that, since $\hat{t} < n/4$, this protocol can easily be adjusted so that polynomials of formal degree $2\hat{t}$ are distributed.

Below we present a security proof for a setting in which none of the initial $2k+1$ coefficients has a fixed value. The security for the case where some of the coefficients are fixed to zero, but only $k+1$ of the coefficients need to remain secret, then follows as a straightforward application of this result.

**Lemma 5.** *Let $F(X,Y)$ be a random symmetric bivariate polynomial of formal degree $\hat{t}$ in each variable and define $f_i(X) := F(X,x_i)$ for $i=1,2,\ldots,n$. If $0 \le d \le \hat{t}$, then any subset of $\hat{t}-d$ polynomials $f_i(X)$ gives no information about the first $d+1$ coefficients of $f(X) := F(X,0)$.*

*Proof.* Assume wlog that the given polynomials are $\{f_i(X)\}_{i=1}^{\hat{t}-d}$. We need to show that for any selection for the first $d+1$ coefficients of $f(X)$, there is a symmetric bivariate polynomial $F(X,Y)$ that is consistent with the given polynomials and the selected coefficients. We show the equivalent statement that there exist symmetric bivariate polynomials $F_j(X,Y)$ for $j=0,1,\ldots,d$ such that $F_j(X,x_i)=0$ for $i=1,2,\ldots,\hat{t}-d$ and all the $d+1$ lower coefficients of $F_j(X,0)$ are zero except for the $j^{\text{th}}$ one, which is equal to one.

By Lemma 3, any selection $c_0,c_1,\ldots,c_d$ for the first $d+1$ coefficients of $f$ leads to a polynomial $f'(X)$ that is consistent with the selection and for which $f'(x_i)=0$ for $i=1,2,\ldots,\hat{t}-d$. Let $C_j$ be the selection where all selected coefficients are zero, except for the first and the $j^{\text{th}}$ one which are equal to one, and let $f_{C_j}$ be the corresponding polynomial with those first coefficients for which $f_{C_j}(x_i)=0$ for $i=1,2,\ldots,\hat{t}-d$.

Define a number of symmetric bivariate polynomials $F_{C_j}(X,Y)$ by setting $F_{C_j}(X,Y) := f_{C_j}(X)f_{C_j}(Y)$ for $j=0,1,\ldots,d$. Then we have that $F_{C_j}(X,x_i) = f_{C_j}(X)f_{C_j}(x_i) = 0$ for $i=1,2,\ldots,\hat{t}-d$ and $F_{C_j}(X,0) = f_{C_j}(X)f_{C_j}(0) = f_{C_j}(X)$. The polynomials $F_0(X,Y) := F_{C_0}(X,Y)$ and $F_j(X,Y) := F_{C_j}(X,Y) - F_{C_0}(X,Y)$ for $j=1,2,\ldots,d$ are now of the desired form. $\square$

We now show that the default zeros in the polynomials $f_i(X)$ that the players receive as their share ensure that the required gap is present in the polynomial $f(X)$.

**Lemma 6.** *Take $x_0=0$. For $i=0,1\ldots,n$, let $f_i(X) := F(X,x_i) = c_{i0} + c_{i1}X + \cdots + c_{i\hat{t}}X^{\hat{t}}$ for certain $c_{ij} \in K$. If $c_{ik}=0$ for at least $\hat{t}+1$ values of $i$, then the coefficient $c_{0k}$ of the polynomial $f_0(X)$ is zero.*

*Proof.* Since $F(X,Y) = \sum_{i,j=0}^{\hat{t}} e_{ij}X^iY^j$, $f_v(X) = \sum_{i=0}^{\hat{t}}(\sum_{j=0}^{\hat{t}} e_{ij}v^j)X^i$ and in particular $f_0(X) = \sum_{i=0}^{\hat{t}} e_{i0}X^i$. Now assume that $c_{i_l k}=0$ for distinct $i_1,i_2,\ldots,i_{\hat{t}+1}$. This amounts to saying that $\sum_{j=0}^{\hat{t}} e_{kj}{i_l}^j = 0$ for $l=1,\ldots,\hat{t}+1$ and therefore the polynomial $\sum_{j=0}^{\hat{t}} e_{kj}Y^j$ has to be the zero polynomial. We conclude that $e_{kj}=0$ for $j=0,1\ldots,\hat{t}$ so that in particular $e_{k0}=c_{0k}=0$. $\square$

### A.2 Multiplication/Resharing Step

Suppose that both $a \in L$ and $b \in L$ have been secret-shared according to the dedicated VSS scheme described above, resulting in distributed polynomials $f_i(X)$

and $g_i(X)$. The aim is to let the players execute a secure resharing protocol that results in a secret-sharing of $ab$ according to the dedicated VSS scheme. The resharing protocol proceeds as follows for every player $P_i$:

1. Player $P_i$ selects a polynomial of the form $v_i(X) = a_ib_i + \sum_{l=2k+1}^{\hat{t}-k} r_l X^l$, where $r_l$ is chosen at random from $K$ for $l = 2k+1, 2k+2, \ldots, \hat{t}-k$ and embeds it in a random symmetric bivariate polynomial $V_i(X,Y) = \sum_{i,j=0}^{\hat{t}} e_{ij} X^i Y^j \in K[X,Y]$ under the restriction that $V_i(X,0) = v_i(X)$ and that $e_{ij} = e_{ji} = 0$ for $j = 1, 2, \ldots, 2k$ and $i = 0, 1, \ldots, n$. This bivariate polynomial is then used for VSS, leading to shared polynomials $v_{ij}(X) := V_i(X, x_j)$.
2. Player $P_i$ selects at random a symmetric bivariate polynomial $R_i(X,Y)$ of formal degree $2\hat{t}-1$ in each variable and distributes using VSS polynomials $r_{ij}(X) := R_i(X, x_j)$, where the evaluations $r_{ij}(0)$ determine the polynomial $r_i(X) := R_i(X,0)$ of formal degree $2\hat{t}-1$.
3. All players $P_j$ broadcast the value $f_j(x_i)g_j(x_i) - v_{ij}(0) + x_j r_{ij}(0)$ and use error correction to reconstruct a polynomial of degree $2\hat{t}$. If the first coefficient of the reconstructed polynomial is not zero, player $P_i$ is disqualified.

First note that $f_j(x_i)g_j(x_i) - v_{ij}(0) + x_j r_{ij}(0) = (f_i g_i - v_i)(x_j) + x_j r_i(x_j)$, so that the players reconstruct the sum of two polynomials where one of the polynomials is random under the restriction that the first coefficient is equal to zero. Since the VSS-schemes have the property that all honest players have consistent shares at the end of the procedure, the polynomials $r_i(X)$, $v_i(X)$, $f_i(X)$ and $g_i(X)$ are uniquely determined when all players pool their shares in these polynomials. Since $\hat{t} < n/4 < n/3$, the same holds for the polynomials $Xr_i(X)$ and $(f_i g_i)(X)$ and therefore also for the polynomial $(f_i g_i - v_i)(X) + Xr_i(X)$. Furthermore, this polynomial has an initial coefficient equal to zero if and only if the first coefficient of $v_i(X)$ is equal to $a_ib_i$. Note also that the additional zero's in the bivariate polynomial $V_i(X,Y)$ ensure to the players that the polynomial $v_i(X)$ is of the proper form.

We need to show that the $n$ polynomials $(f_i g_i - v_i)(X) + Xr_i(X)$ together with $t$ evaluations on the points $x_1, x_2, \ldots, x_t$ for every polynomial $r$, $v_i$, $f_i$ and $g_i$ do not give any information about $a$, $b$ or $ab$. First, we can conclude by the following lemma that the sum of two arbitrary polynomials of degree $2\hat{t}$ together with $t$ evaluations for these polynomials give no information about the first $\hat{t} - t + 1$ first coefficients of one of these two polynomials.

**Lemma 7.** *Let $f$ and $g$ be polynomials of formal degree $\hat{t}$ and let the polynomial $f + g$ and evaluations $f(x_i)$ and $g(x_i)$ be given for $i = 1, 2, \ldots, d$. Then $f + g$ together with the given evaluations $f(x_i)$, $g(x_i)$ give no information about the first $\hat{t} - d + 1$ coefficients of $f$.*

*Proof.* By Lemma 3, for any selection $C = (c_0, c_1, \ldots, c_{\hat{t}-d+1})$ there exists a polynomial with these values as the first $\hat{t}-d+1$ coefficients that evaluates to zero in the points $x_1, x_2, \ldots, x_d$. Then adding this polynomial to $f$ and subtracting it from $g$ leads to consistent polynomials $f'$ and $g'$ with different initial coefficients, while the sum $f' + g'$ remains the same. This works for every arbitrary selection $C$, and therefore the given information is consistent with any selection for the first coefficients of $f$. $\square$

As a consequence of the lemma, given the evaluations of $t$ players we can choose polynomials of formal degree $\hat{t}$ with arbitrary first $k+1$ coefficients that evaluate to zero in the given points and add them to the polynomials $f_i$ and $g_i$ to give polynomials $f_i'$ and $g_i'$. Then, the polynomial $f_i'g_i' - f_ig_i$ can be subtracted from $r_i$, which gives a polynomial $r_i'$ that is consistent with the given points on $r_i$, but for which the sum $f_i'g_i' + r_i'$ is equal to $f_ig_i + r_i$. Therefore, no information about $a$, $b$ or $ab$ is leaked during the protocol.

### A.3 Local Computation

In order to obtain the desired polynomials $h_i(X)$, every player $P_i$ now locally computes the polynomial

$$h_i(X) = \sum_{j=1}^{n} \left( \sum_{l=0}^{k} \psi_k(\epsilon_i)(X^l + x_i{}^l) \right) v_{ji}(X),$$

where $\psi_l : K^n \to K$ for $l = 1, 2, \ldots, n$ have been defined in Section 5.

Define $h(X) := \sum_{i=1}^{n}(\sum_{l=0}^{k} \psi_l(\epsilon_i)X^l)v_i(X)$. Then it is easy to verify that $h(X)$ has degree $\hat{t}$ and we can write it in the form

$$\left( \sum_{l=0}^{k} \left( \sum_{i=1}^{n} \psi_l(\epsilon_i) a_i b_i \right) X^l \right) + \sum_{l=2k+1}^{\hat{t}} r_l'' X^l$$

for certain $r_{2k+1}'', r_{2k+2}'', \ldots, r_{\hat{t}}'' \in K$. In particular, the first $k+1$ coefficients are the coefficients of $ab$. Below, we show that the evaluations $h_i(0)$ all give evaluations on $h(X)$ and that for all $i, j \in \{1, 2, \ldots, n\}$ we have that $h_i(x_j) = h_j(x_i)$, so that there exists a symmetric bivariate polynomial $H(X, Y)$ such that $H(X, 0) = h(X)$ and $H(X, x_i) = h_i(X)$. Therefore, the resulting sharing is of the desired form.

The following two, easy to verify lemmas show that the polynomials $h_i(X)$ that the players obtain are part of a proper sharing of the polynomial $h(X)$. Therefore, the protocol described above gives us a proper sharing of the new (product) secret.

**Lemma 8.** $\forall 1 \leq i \leq n : h_i(0) = h(i)$.

**Lemma 9.** $\forall 1 \leq i, j \leq n : h_i(j) = h_j(i)$.