Fair and Privacy-Preserving Multi-Party Protocols for Reconciling Ordered Input Sets

(Extended version)

Georg Neugebauer* and Ulrike Meyer* and Susanne Wetzel†

*UMIC Research Center, RWTH Aachen University, Germany

{neugebauer, meyer}@umic.rwth-aachen.de

†Stevens Institute of Technology, Hoboken, New Jersey 07030

swetzel@stevens.edu

April 20, 2011

In this paper, we introduce the first protocols for multi-party, privacy-preserving, fair reconciliation of ordered sets. Our contributions are twofold. First, we show that it is possible to extend the round-based construction for fair, two-party privacy-preserving reconciliation of ordered sets to multiple parties using a multi-party privacy-preserving set intersection protocol. Second, we propose new constructions for fair, multi-party, privacy-preserving reconciliation of ordered sets based on multiset operations. We prove that all our protocols are privacy-preserving in the semi-honest model. We furthermore provide a detailed performance analysis of our new protocols and show that the constructions based on multisets generally outperform the round-based approach.

Keywords privacy, secure group computation, cryptographic protocols, private set intersection, multi-party protocols

1. Introduction

Recently, protocols were proposed that allow two parties to reconcile their ordered input sets in a privacy-preserving and fair manner [13]. Fair reconciliation of ordered sets is defined as a protocol that allows two parties whose input sets are ordered according to their individual preferences to determine those inputs the parties have in common that additionally maximize a specific combined preference order. Many applications of such fair, privacy-preserving reconciliation protocols exist and range from simple scheduling applications to the reconciliation of policies in Future Internet architectures [13]. Specifically, this work will enable a fair and privacy-preserving version of Doodle [4]. Currently, Doodle allows several parties to schedule a meeting in a distributed and efficient manner. However, today's Doodle application does not allow the parties to order the time slots at which they would be available for the meeting according to their preferences. Furthermore, all parties see which time slots were (not) selected by the others. In this context, our work allows an extension of Doodle which will take the preferences of all parties into account when determining the best time slot for the meeting. In addition, the advanced Doodle will keep the settings of all parties private. I.e., neither the information on what time slots were (not) selected nor a party's corresponding preferences will be disclosed to the others.

The protocols introduced in [13] are designed for two parties. The maximizing of the parties' individual preferences is achieved by carrying out privacy-preserving set intersection protocols on specifically chosen input sets in a particular order. This order directly corresponds to the combined preference order itself. In this context, our contributions in this paper are twofold. We first show that it is possible to extend the round-based construction of [13] to the multi-party case using the multi-party private set intersection protocol introduced in [10]. Furthermore, we propose a new more efficient construction for fair, multi-party, privacy-preserving reconciliation protocols of ordered sets. The core of the new construction is an intricate encoding of both the parties' input sets and their associated preferences. This encoding is based on multisets. The new protocols integrate the intersection, union, and element reduction operations on multisets, which were first introduced in [10]. We prove that the protocols for both of our constructions are privacy-preserving in the semi-honest model. In addition, we provide a performance analysis of our new protocols with respect to communication and computation overhead and show that for more than six parties the construction based on multisets outperforms the round-based approach.

This is the extended version of the paper presented at the ISC 2010 [15]. This version contains more details on the protocol description, a detailed performance analysis, a generalization of Lemma 2 [10] and a proof of the generalized Lemma 2. The remainder of this paper is organized as follows: In Section 2 we discuss related work. Section 3 briefly reviews basic components used in our constructions. Section 4 details our new round-based and multiset-based constructions. In Section 5 we provide a performance analysis of the two new constructions. We close the paper with some remarks on ongoing and future work.

2. Related Work

The basis of work for this paper is preference-maximizing privacy-preserving reconciliation of ordered sets. Two-party protocols for this type of operation were introduced in [12]. [13] further develops these protocols and shows that it is possible to construct two-party protocols that are privacy-preserving in the semi-honest model from any privacy-preserving set intersection protocol such as [1, 3, 7, 8, 9, 5, 17]. The two-party protocols described in [12, 13] use several rounds of computing the intersection of input sets of the two parties. The input sets in these rounds are chosen such that upon termination of the protocol only one preference-maximizing common set element is revealed to each of the parties. The detailed protocol descriptions in [12, 13] are based on [5] which uses oblivious polynomial evaluation for computing the intersection of two private datasets in a privacy-preserving manner.

Compared to [12, 13] our main contribution is the generalization of the protocols to multiple parties. Specifically, we suggest two new constructions that achieve this generalization. The first one leads to a round-based construction that uses a multi-party private set intersection protocol such as [10, 11, 14, 16] in each round. We describe this round-based construction and analyze its performance in terms of the total number of runs of the multi-party private set intersection protocol required. In addition, we show that our multi-party round-based constructions are privacy-preserving in the semi-honest model.

The second multi-party construction introduced in this paper makes use of the results on private multiset operations and protocols introduced in [10]. Here, multisets refers to sets in which elements may occur more than once. The authors of [10] specify algorithms for privacy-preserving operations not only for the intersection of multisets but also for the union of multisets, and for element reduction. In addition, they show that based on these operations, any function over multisets that can be expressed by the grammar

$$\Upsilon ::= S_i \mid Rd_t(\Upsilon) \mid \Upsilon \cap \Upsilon | S_i \cup \Upsilon | \Upsilon \cup S_i \tag{1}$$

can be computed in a privacy-preserving manner. Furthermore, the authors describe protocols for (cardinality) set intersection ($S_1 \cap ... \cap S_n$) and different forms of threshold set union (Rd_t ($S_1 \cup ... \cup S_n$)). They prove the security of their protocols in the semi-honest as well as the malicious model [10]. In addition, they analyze the communication overhead of their protocols.

Compared to [10] the main contribution of this paper is the idea to encode the rank of an element in an ordered input set such that the rank corresponds to the number of occurrences of that element in a corresponding multiset and to show that the preference-maximizing objectives can be expressed in the above grammar. For these particular functions no detailed description of a privacy-preserving protocol is provided in [10]. We therefore specify these new multi-party protocols in detail. In addition, we provide a detailed analysis of both the communication and the computation overhead of our new protocols and show that they are privacy-preserving in the semi-honest model.

3. Preliminaries

3.1. Ordered Sets and Preferences

Throughout this paper, we consider n parties $P_1, ..., P_n$ with input sets $R_1, ..., R_n$ chosen from a common domain R. Each input is a set of k elements, $r_{i1}, ..., r_{ik}$. Each element r_{ij} of party P_i is represented as a bit-string of length m. Furthermore, we assume that each party can totally order the elements in its input set R_i according to its preferences. The rank of an element r_{ij} is determined by $rank_{P_i}(r_{ij}) = k - j + 1$ ($j = 1, ..., k$) which is a bijective function that induces a total order $\leq_{P_i}$ on R_i. The most preferred element has the highest rank. The goal of the parties is to not only determine the elements they have in common but determine those shared elements which maximize their combined preferences. Analogously to the definition in [13], we define a preference order composition scheme for n parties as follows:

Definition 3.1 *For each party P_i with $i = 1, ..., n$, let $\leq_{P_i}$ be the preference order induced on the input set R_i denoted by $rank_{P_i}$. A combined preference order $\leq_{\{P_1,...,P_n\}}$ is a total pre-order induced on the intersection of the parties' inputs $R_1 \cap ... \cap R_n$ by a real-valued function f—in the sequel referred to as preference order composition scheme.*

In the remainder of this paper we focus on two specific preference order composition schemes, namely the *minimum of ranks* and the *sum of ranks* composition scheme.

Definition 3.2 *The minimum of ranks composition scheme is defined by the real-valued function $f(x) = min\{rank_{P_1}(x), ..., rank_{P_n}(x)\}$ for $x \in R_1 \cap ... \cap R_n$.*

Definition 3.3 *The sum of ranks composition scheme is defined by the real-valued function $f(x) = rank_{P_1}(x) + ... + rank_{P_n}(x)$ for $x \in R_1 \cap ... \cap R_n$.*

In the Doodle application, the parties' input sets would consist of (possible) time slots (e.g., a date associated with a time) which are additionally ordered according to the parties' preferences.

Using the sum of ranks composition scheme will then yield a common time slot which is in total most preferred among all participating parties. The minimum of ranks composition scheme will yield a common time slot in which the least assigned preference is maximized.

3.2. Homomorphic Cryptosystem

Our protocols require a threshold version of a semantically secure, additively homomorphic, asymmetric cryptosystem.

Additively Homomorphic. Given the ciphertexts $c_1 = E(m_1)$ and $c_2 = E(m_2)$, the encryption of the sum of the plaintexts $E(m_1 + m_2)$ can be determined with an operation $+_h$ in the ciphertext domain knowing only the ciphertexts c_1 and c_2 as $E(m_1 + m_2) = c_1 +_h c_2$. This can be further generalized. Given a ciphertext $c = E(m)$ and a scalar value s, the encryption of the product $m \cdot s$ can be determined by applying the operation $\times_h$ s times in the ciphertext domain using only the ciphertext c as $E(m \cdot s) = c \times_h s = c +_h \ldots +_h c$.

Threshold Decryption. For an (n, n)-threshold cryptosystem with n parties, the private key k_{priv} is shared among the n parties with each party P_i holding a private share s_i $(1 \leq i \leq n)$. Given a ciphertext $c = E(m)$, the n parties must cooperate in order to decrypt the ciphertext c. The threshold decryption of c requires each party P_i to use its private share s_i of the private key k_{priv}. One example for a suitable system that is semantically secure and additively homomorphic, is the threshold version of the Paillier cryptosystem [2].

3.3. Prior Results On Two-Party Preference-Maximizing Protocols

The protocols in [13] introduce fairness in the reconciliation process of two ordered input sets for the two preference composition schemes defined above. Each set element is associated with a preference that corresponds to its rank in the ordered set. Upon protocol termination, both parties have learned nothing but the set element that maximizes the combined preference order. The protocol works as follows.

The protocol consists of multiple rounds. In each round, all pairs of set elements are compared according to the combined preference order. For the sum of ranks (minimum of ranks) composition scheme, there are up to $2n - 1$ (respectively n) rounds. The order in which the set elements are compared guarantees that if a match is found, it is the maximum according to the chosen combined preference order. In each round, a set element is interpreted as a root of a polynomial. The set intersection is then calculated in a privacy-preserving manner using oblivious polynomial evaluation as introduced by Freedman et al. [5]. It is furthermore shown in [5] that the protocols can be generalized to use any arbitrary two-party privacy-preserving set intersection protocol.

3.4. Prior Results On Multiset Operations

Kissner and Song [10] specify privacy-preserving algorithms for the computation for three operations on multisets: intersection ($\cap$), union ($\cup$), and element reduction by t (Rd_t). In addition, they show that based on these operations, any function over multisets that can be expressed by the grammar in equation (1) can be computed in a privacy-preserving manner. Here, S_i is a multiset of a participating party P_i and $t \geq 1$. Note that the union operation can only be computed if one of the two operands is known to some party P_i.

The multisets $S_i = \{s_{i1}, ..., s_{ik}\}$ are represented as polynomials $f(X) = \prod_{j=1}^{k}(X - s_{ij})$. I.e., an element appearing y times in the multiset S_i is a y-fold root of the corresponding polynomial f.

3.4.1. Polynomial Operations

The operations union, intersection, and element reduction on multisets in [10] are based on operations on the polynomials representing them.

Union. The union $S_1 \cup S_2$ of two multisets S_1 and S_2 (represented by polynomials f_1 and f_2 respectively) can be expressed by the multiplication of the polynomials as $f_1 * f_2$. Each element a appearing y_1 times in S_1 and y_2 times in S_2 with $y_1, y_2 \geq 0$ occurs $y_1 + y_2$ times in the resulting multiset.

Intersection. The intersection $S_1 \cap S_2$ of two multisets S_1 and S_2 (represented by the polynomials f_1 and f_2 of degree d respectively) can be expressed by the polynomial $f_1 * r + f_2 * s$, where r, s are random polynomials of degree d. Each element a appearing y_1 times in S_1 and y_2 times in S_2 with $y_1, y_2 > 0$ occurs $min\{y_1, y_2\}$ times in the resulting multiset.

Element Reduction. The reduction $Rd_t(S)$ (by t) of a multiset S represented by polynomial f can be expressed by the polynomial $\sum_{j=0}^{t} f^{(j)} * F_j * r_j$, where $f^{(j)}$ is the j-th derivative of f and r_j, F_j are random polynomials of degree $deg(f^{(j)})$. F_j is chosen such that no roots of F_j are elements of the overall domain R. Each element a occurring y times in S with $y \geq 0$ occurs $max\{y-t, 0\}$ times in the resulting multiset. The correctness of these polynomial representations is proven in [10]. Additionally, the authors in [10] show that one cannot learn more information about the initial multisets observing the result of the operations on polynomials than what can be deduced from the result of applying the operations on the multisets directly.

3.4.2. Encrypted Polynomial Operations

As shown in Kissner and Song, Section 4.2.2 [10], assuming a semantically secure homomorphic encryption function E, it is possible to compute the sum of two encrypted polynomials, the derivative of an encrypted polynomial, and the product of an unencrypted polynomial and an encrypted polynomial without knowledge of the plaintext coefficients of the encrypted polynomials. In particular, let f denote a polynomial of degree d represented by its coefficients $f[0], \ldots, f[d]$, and let $E(f)$ denote the encrypted polynomial with encrypted coefficients $E(f[0]), ..., E(f[d])$.

Sum of Encrypted Polynomials: Given encryptions $E(f_1)$, $E(f_2)$ of two polynomials f_1, f_2 of degree d_1, d_2, the encryption of the polynomial $g = \ f_1 + f_2$ can be computed as

$$E(g[i]) = E(f_1[i]) +_h E(f_2[i]) \text{ where } 0 \leq i \leq \max\{d_1, d_2\}. \tag{2}$$

I.e., computing the encryption of the sum requires $(\max\{d_1, d_2\} + 1)$ $+_h$-operations.

Derivative of an Encrypted Polynomial: Given the encryption $E(f)$ of a polynomial f of degree d, the encryption of polynomial $g = \frac{d}{dx} f$ can be determined as

$$E(g[i]) = (i + 1) \times_h E(f[i + 1]) \text{ where } 0 \leq i \leq d - 1. \tag{3}$$

I.e., computing the encryption of the derivative requires d $\times_h$-operations.

Product of an Unencrypted Polynomial and an Encrypted Polynomial: Given the encryption $E(f_1)$ of a polynomial f_1 of degree d_1 and a polynomial f_2 of degree d_2, the encryption of $g = f_1 * f_2$ can be determined as

$$E(g[i]) = (f_2[0] \times_h E(f_1[i])) +_h (f_2[1] \times_h E(f_1[i-1])) +_h \\ \dots +_h (f_2[i] \times_h E(f_1[0])) \text{ where } 0 \leq i \leq d_1 + d_2.$$

Note that $E(f_1) = E(f[0]), \dots, E(f[d_1])$, $f_2 = f[0], \dots, f[d_2]$ and undefined array positions are treated as zero, i. e. that e. g. $E(f_1[d_1 + d_2]) = 0$ or $f_2[d_1 + d_2] = 0$. An upper bound on the number of operations necessary to compute the product is as follows:

$$+_h : \quad \sum_{i=0}^{d_1+d_2} i = \frac{(d_1 + d_2) \cdot (d_1 + d_2 + 1)}{2} \tag{4}$$

$$\times_h : \quad \sum_{i=0}^{d_1+d_2} i + 1 = \sum_{i=1}^{d_1+d_2+1} i = \frac{(d_1 + d_2 + 1) \cdot (d_1 + d_2 + 2)}{2} \tag{5}$$

3.5. Adversary Model

In this paper, we consider the honest-but-curious adversary model, which is also referred to as the *semi-honest model* [6]. In the semi-honest model all parties act according to the prescribed actions in the protocols. They may, however, try to infer as much information as possible from all results obtained during the execution of the protocol. Consequently, a protocol is said to be privacy-preserving in the semi-honest model, if no party gains any information about the other party's private input other than what can be deduced from the output of the protocol and its own private input. While not the strongest model possible, the semi-honest model is suitable for all applications in which none of the parties is more ill-intended than just being curious. For example, in the advanced Doodle application where the parties are simply interested in finding a common time for a meeting, it is reasonable to assume that none of the parties has any purely malicious intend.

4. Multi-Party Privacy-Preserving Reconciliation of Ordered Sets

We start with a description of our new protocols for privacy-preserving, preference-maximizing reconciliation of ordered sets for two or more parties. We start with a more formal definition of such reconciliation protocols.

Definition 4.1 *A privacy-preserving, preference-maximizing protocol for a preference order composition scheme C is a multi-party protocol between n parties $P_1, \dots, P_n$ with inputs $R_1, \dots, R_n$ each containing k elements drawn from the same domain R and preference orders $\leq_{P_1}, \dots, \leq_{P_n}$. Upon completion of the protocol, no party learns anything about any other party's inputs and preferences but what can be deduced from the elements that maximize the combined preference order $\leq_{\{P_1,\dots,P_n\}}$ and their respective ranks under $\leq_{\{P_1,\dots,P_n\}}$.*

In the following, we focus on the combined preference order composition schemes minimum of ranks and sum of ranks (see Section 3.1). We first present a new protocol (for both preference order composition schemes) which generalizes the round-based construction (see Section 3.3) for multiple parties. Then, we detail our new multiset-based construction for multi-party privacy-preserving reconciliation of ordered sets. This eliminates the need for a round-based proceeding thus resulting in a substantial improvement in efficiency in the case of more than six parties.

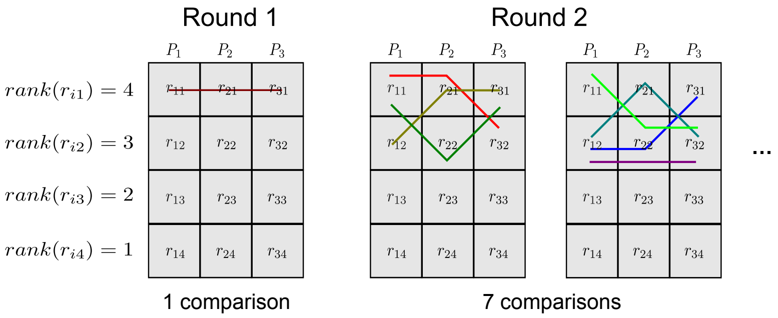

Figure 1: Minimum of ranks composition scheme with three parties and four input elements for the first two rounds. In Round 1, the minimum of ranks is four. In Round 2, the minimum of ranks is three which results in seven different input combinations.

4.1. Round-based Constructions

The main idea in generalizing the round-based privacy-preserving reconciliation of ordered sets construction is to use the multi-party set intersection protocol Set-Intersection-HbC (see Section 3.4) to compute the intersection of one-element subsets of each party's inputs. This is done in such a way that the order in which the intersections are computed ensures that the first non-empty intersection found contains the element that maximizes the combined preference order under the respective preference order composition scheme.

4.1.1. Minimum of Ranks Composition Scheme

The n parties participate in a multi-round protocol where each round consists of one or more executions of a multi-party set intersection protocol with varying inputs. In the first round, party P_i's input set contains its most preferred element r_{i1}. In each of the following rounds $s = 2, \dots, k$, the parties participate in $s^n - (s-1)^n$ protocol runs of Set-Intersection-HbC. Each run of Set-Intersection-HbC takes a one-element subset of the inputs of each party as input. The order in which the parties select the inputs is determined such that in round s the maximum of the minimum of ranks of all of the input sets is $k - s + 1$ (see Figure 1 for details). If the result of Set-Intersection-HbC is the empty set, then the parties proceed with the next input values. Otherwise, the protocol terminates and each party learns the preferred element according to the maximum of the minimum of ranks composition scheme and the round in which the match was found. Since the input sets to each run of the multi-party set intersection only contain one element, the input sets can be represented as polynomials of degree one. As a consequence, the performance of the round-based approach is dominated by the number of runs of Set-Intersection-HbC. In the worst-case, i.e., if the parties do not share any element, all possible combinations of the k input elements of the n parties result in the execution of Set-Intersection-HbC. Therefore, in the worst-case the overall number of executions of Set-Intersection-HbC is k^n. The communication complexity of one run of the Set-Intersection-HbC protocol with multisets of size one is $O(c \cdot n)$ (c is the number of colluding attackers) [10]. The same holds for the computation complexity. Thus, in the worst-case, the overall communication and computation complexity of the round-based multi-party privacy-preserving reconciliation of ordered sets approach is $O(c \cdot n \cdot k^n)$.

4.1.2. Sum of Ranks Composition Scheme

Analogous to the minimum of ranks composition scheme, it is possible to generalize the round-based reconciliation protocols for multiple parties. The multi-party protocol requires at most $(n{\cdot}k)-(n-1)$ rounds. Each round entails a number of runs of Set-Intersection-HbC. Specifically, in round s, the inputs r_{ij_i} with $i = 1, ..., n$ for the different runs of Set-Intersection-HbC are determined such that $s = \sum_{i=1}^{n} rank_{P_i}(r_{ij_i})$ where the j_i are chosen from $\{1, ..., k\}$. Again, as long as the result of Set-Intersection-HbC is the empty set, the parties proceed with the next input values. Otherwise, the protocol terminates and each party learns the preferred policy rule according to the maximum of the sum of ranks composition scheme and the round in which the match was found. As the maximum number of runs of Set-Intersection-HbC is again equal to k^n, the overall communication and computation complexities of the round-based multi-party, privacy-preserving reconciliation of ordered sets approach for the sum of ranks composition scheme are $O(c \cdot n \cdot k^n)$ in the worst-case.

4.1.3. Some Remarks

Combining the results in [13] and [10] directly provides for the new protocols being privacy-preserving in the semi-honest model. It is important to note that based on the constructions described in Section 3.3, it is possible to achieve a slightly stronger privacy guarantee than that of Definition 4.1. This is due to the fact that the protocols are designed to abort as soon as the first match is found. The protocols could be easily modified to output all maximizing rules found in one round. Furthermore, the Set-Intersection-HbC protocol used in both constructions above may be replaced with any other privacy-preserving, multi-party set intersection protocol.

4.2. Multiset-based Constructions

It would be nice if there were a more efficient solution than the round-based construction detailed above. The idea was to find a way to encode preferences as part of the inputs directly (instead of in the order in which the parties compare their inputs). It turns out that a good candidate for this approach is using multisets and encoding the rank of an input as the number of times which the input occurs in the multiset. Now the question is if one can find a way to represent the inner workings of the preference order composition scheme in the powerful grammar for privacy-preserving set operations proposed in [10]—i.e., if one can find a way to express maximizing the preferences according to the preference composition scheme in question by means of intersection, union, and reduction operations. As we will see, expressing the minimum of ranks composition scheme in this grammar is surprisingly easy, while expressing the sum of ranks composition scheme is by far not that straightforward. In particular, we show that our constructions yield functions which are not supported by prior developed protocols (see Section 3.4.) and thus require the design of new protocols.

Analogous to the work in [10], we assume that at most $c < n$ of the n parties collude. Furthermore, (k_{pub}, k_{priv}) denotes a public/private key pair for the asymmetric, semantically secure, homomorphic threshold cryptosystem. Each party P_i holds an authentic copy of the public key k_{pub} as well as its share s_i $(1 \leq i \leq n)$ of the private key k_{priv}. The shares $s_1, \ldots, s_n$ are generated by means of a suitable (n, n)-secret sharing scheme, e. g. [18].

4.2.1. Minimum of Ranks Composition Scheme

For each party P_i ($i = 1, \dots, n$) we represent its input elements together with their respective ranks as the multiset

$$S_i := \{\underbrace{(r_{i1}), \dots, (r_{i1})}_{k\ times}, \underbrace{(r_{i2}), \dots, (r_{i2})}_{k-1\ times}, \dots, \underbrace{(r_{i(k-1)}), (r_{i(k-1)})}_{twice}, \underbrace{(r_{ik})}_{once}\}.$$

I.e., each element occurs in S_i as often as its rank indicates such that the most preferred element r_{i1} occurs k times while the least preferred element r_{ik} occurs only once. Using the grammar and basic operations described in Section 3.4, we now show that if it is possible to compute

$$Rd_t(S_1 \cap ... \cap S_n) \tag{6}$$

(where $k > t \geq 0$ is an appropriate reduction value) in a privacy-preserving manner, then this yields a multi-party, privacy-preserving reconciliation protocol of ordered sets for the minimum of ranks composition scheme – also referred to as MPROS^{MR}. Intuitively speaking, for the computation of Equation (6) the protocol entails the following steps:

1. Each party determines the polynomial representation of its multiset S_i.

2. All parties calculate the set intersection on input sets $S_1, ..., S_n$. After this step, all parties hold an encrypted polynomial that represents $S_1 \cap ... \cap S_n$. This intersection encodes not only the elements which the parties have in common but also the respective minimum preference for each of these elements over all the participants.

3. All parties iteratively calculate the element reduction by t. The first time this step is executed the reduction value $t = k - 1$ is used. The reduction operation is applied on the result of the set intersection from the previous step. The goal of the reduction step is to determine the rule in the intersection $S_1 \cap ... \cap S_n$ for which the minimum preference is maximized, i.e., to perform element reduction using the largest possible t. A reduction by t eliminates up to t occurrences of each unique element in the multiset $S_1 \cap ... \cap S_n$. If t is too large, the reduction will output the encryption of a polynomial representing the empty set.

4. All parties participate in the threshold decryption of the result of the previous step. Specifically, each party checks whether at least one of its input elements is a root of the polynomial computed as part of the previous step. This will not be the case until the previous step results in a non-empty set which in turn corresponds to the maximum of the minimum of ranks. As long as the previous step yields an empty set, the parties iterate Steps 3 and 4 with a decreasing value of t.

Figure 2 details each of the steps of this new protocol. The n parties $P_1, ..., P_n$ are arranged in a ring structure. To increase readability, we generally omit the necessary mod n in the protocol description. For example, the P_ℓ for an arbitrary ℓ is in fact the shorthand for $P_{\ell \bmod n}$.

In Step 1, all parties construct polynomials which represent their totally ordered input elements. Then, they calculate the encryption of their polynomial (see Section 3.4.2) and send it to the next c parties. Next, each party chooses $c + 1$ random polynomials and calculates the encryption of the scalar product ϕ_i of the c received encrypted polynomials, its own, and the randomly chosen polynomials using the algorithms given in Section 3.4.2. Each ϕ_i represents some part of the polynomial which in turn encodes the set intersection.

There are n parties $P_1, ..., P_n$ of which at most $c < n$ collude as attackers. $F_0, ..., F_t$ are fixed polynomials of degree $0, ..., k-1$ that do not have any elements of the overall domain R as root.

Set Intersection

1. Each party P_i $(i = 1, ..., n)$
   a) calculates the polynomial $f_i(X) = (X - r_{i1})^k \cdot (X - r_{i2})^{k-1} \cdot ... \cdot (X - r_{ik})^1$ where $deg(f_i(X)) = \frac{k \cdot (k+1)}{2}$,
   b) sends $E(f_i(X))$ to parties $P_{i+1}, ..., P_{i+c}$,
   c) chooses $c+1$ random polynomials $q_{i,0}, ..., q_{i,c}$ of degree $deg(f_i(X))$,
   d) calculates $E(\phi_i)$ with $\phi_i = f_{i-c} * q_{i,c} + ... + f_i * q_{i,0}$.
2. Party P_1 sends the encryption of $\lambda_1 = \phi_1$ to party P_2.
3. For each $i = 2$ to n each party P_i:
   a) receives $E(\lambda_{i-1})$ from party P_{i-1},
   b) calculates the encryption of $\lambda_i = \lambda_{i-1} + \phi_i$,
   c) sends $E(\lambda_i)$ to party P_{i+1}.
4. Party P_1 distributes $E(\lambda_n)$ with $p = \lambda_n = \sum_{i=1}^{n} f_i * (\sum_{j=0}^{c} q_{i+j,j})$ to $c+1$ randomly chosen parties $P_{j_1}, \ldots, P_{j_{c+1}}$ where $deg(\lambda_n) = k \cdot (k+1)$.

Reduction Step (t=k-1,...,0)

5. Each party P_{j_s} $(s = 1, ..., c+1)$
   a) calculates the encryption of the $1, ..., t$-th derivatives of polynomial p, denoted by $p^{(1)}, ..., p^{(t)}$,
   b) chooses $t+1$ random polynomials $q'_{s,0}, ..., q'_{s,t}$ of degree $0, ..., t$,
   c) calculates the encryption of the polynomial $p^*_s = \sum_{l=0}^{t} p^{(l)} * F_l * q'_{s,l}$ and sends it to all other parties.

Decryption

6. All parties
   a) compute the encryption of the sum Φ of the polynomials p^*_s, $s = 1, \ldots, c+1$,
   b) perform a threshold decryption to obtain the polynomial $\Phi = \sum_{s=1}^{c+1} p^*_s = \sum_{l=0}^{t} p^{(l)} * F_l * (\sum_{s=1}^{c+1} q'_{s,l})$ where $deg(\Phi) = k \cdot (k+1) + t$
7. Each party P_i $(i = 1, ..., n)$ checks for each of its input elements $r_{il} \in S_i$ $(l = 1, ..., k)$ whether it is an element of $Rd_t(S_1 \cap ... \cap S_n)$ by checking whether r_{il} is a root of Φ, i.e., whether $(X - r_{il})|\Phi$. If no match is found, proceed with Step 5b) and a decreasing value of t.

Figure 2: Protocol for the minimum of ranks composition scheme

In Steps 2 and 3, the ϕ_i's are combined in a circular way starting with party P_1 sending $E(\lambda_1) = E(\phi_1)$ to party P_2 (Step 2). Upon receiving $E(\lambda_{i-1})$ from party P_{i-1}, party P_i $(i = 2, \ldots, n)$ then calculates the encryption of $\lambda_{i-1} + \phi_i$ and sends it to P_{i+1} using the sum operation on encrypted polynomials.

In Step 4, P_1 distributes the result $p = \lambda_n$ to $c+1$ randomly chosen other parties. This ensures that at least one honest party receives p.

Note that in the resulting polynomial p, each contributing polynomial f_i is blinded by $c+1$ random values. Consequently, the corresponding sum $\sum_{j=0}^{c} q_{i+j,j}$ is uniformly distributed and up to c colluding attackers cannot deduce any information from it.

Step 5 corresponds to the element reduction by t. According to Section 3.4.2, the $c+1$ parties calculate the encryption of the $1, ..., t$-th derivatives of p and construct the encryption of the polynomial p^*_s which corresponds to the element reduction by t of the multiset represented by polynomial p. The result p^*_s is broadcast to all parties.

In Step 6, all parties perform a threshold decryption to obtain polynomial Φ. Again, each polynomial $p^{(l)}$ in Φ is blinded by $c+1$ random values such that the sum is uniformly distributed and up to c colluding attackers may not infer any information from it.

In Step 7, each party checks if the multiset represented by the polynomial Φ contains one of its input elements. If this is the case, the protocol terminates and the elements that maximize the combined preferences of all parties are found. If this is not the case, the parties reduce the value t by one and repeat Steps 5b-7.

Correctness and Privacy. The above protocol first executes the multiset intersection operation followed by a sequence of element reductions. The multisets S_i of each party P_i are constructed in a way such that the intersection of the multisets leads to a multiset in which each common element r of the n parties occurs exactly $\min\{\text{rank}_{P_1}(r), \ldots, \text{rank}_{P_n}(r)\}$ times. Obviously, the maximum of the minimum of ranks of the common elements is between k and 1. As a consequence, if the element reduction by $k-1$ of the multiset resulting from the intersection leads to a non-empty set of common input elements, then the rank of these elements maximizes the minimum of ranks. If the reduction by $k-1$ leads to an empty set, the maximum of the minimum of ranks is lower than k. Continuing this argument with an iterative reduction by $k-2$,...,0 proves the correctness of our construction. In our protocol, each party P_i learns all maximally preferred common elements[1] and the value t corresponding to the reduction in which these rules were found[2].

In order to prove that the new protocol is privacy-preserving in the semi-honest model, we can directly build on the results by Kissner et al. [10]. They show that in the presence of at most c colluding attackers, their multiset operations (union, intersection, and element reduction) are privacy-preserving in the semi-honest model and that these operations can be arbitrarily composed in a privacy-preserving manner. As our protocol combines the intersection with several element reduction operations, it is privacy-preserving in the semi-honest model in the presence of at most c attackers as well. A detailed performance analysis of the protocol can be found in Section 5 and Appendix 5.1.

4.2.2. Sum of Ranks Composition Scheme

We once again represent the input elements of each party P_i $(i = 1, \ldots, n)$ together with their respective ranks as the multiset

$$S_i := \{\underbrace{(r_{i1}), \ldots, (r_{i1})}_{k\ times}, \underbrace{(r_{i2}), \ldots, (r_{i2})}_{k-1\ times}, \ldots, \underbrace{(r_{i(k-1)}), (r_{i(k-1)})}_{twice}, \underbrace{(r_{ik})}_{once}\}.$$

In addition, we define

$$S_i' = \{\underbrace{(r_{i1}), \ldots, (r_{i1})}_{n \cdot k\ times}, \underbrace{(r_{i2}), \ldots, (r_{i2})}_{n \cdot k\ times}, \ldots, \underbrace{(r_{ik}), \ldots, (r_{ik})}_{n \cdot k\ times}\}.$$

Using the grammar and basic operations described in Section 3.4, we now show that if it is possible to compute

$$Rd_t((S_1 \cup \ldots \cup S_n) \cap S_1' \cap \ldots \cap S_n') \tag{7}$$

in a privacy-preserving manner, then this yields a multi-party privacy-preserving reconciliation protocol of ordered sets for the sum of ranks composition scheme − also referred to as MPROS^{SR}. It is also important to note that the seemingly simpler construction $Rd_t(S_1 \cup \ldots \cup S_n)$

[1] Note that in the special case where all parties hold the same input set but all in complementary order, the maximally preferred common elements are the complete common input set.

[2] As a reminder, the round-based protocol building on the original construction described in Section 3.3 can be modified such that party P_i also learns all maximally preferred common elements and not just one of them.

There are n parties $P_1, ..., P_n$ of which at most $c < n$ collude as attackers. $F_0, ..., F_t$ are fixed polynomials of degree $0, ..., nk-1$ that do not have any elements of the overall domain R as root.

Set union

1. Each party P_i $(i = 1, ..., n)$ calculates the polynomial
   $f_i(X) = (X - r_{i1})^k \cdot (X - r_{i2})^{k-1} \cdot ... \cdot (X - r_{ik})^1$ with $deg(f_i(X)) = \frac{k \cdot (k+1)}{2}$.
2. Party P_1 sends the encryption of $\delta_1 = f_1$ to party P_2.
3. For each $i = 2$ to n each party P_i $(i = 2, ..., n)$
   a) receives $E(\delta_{i-1})$ from party P_{i-1},
   b) calculates the encryption of $\delta_i = \delta_{i-1} * f_i$,
   c) sends $E(\delta_i)$ to party P_{i+1}.
4. Party P_1 distributes $E(\delta_n)$ with $p_1 = \delta_n = \prod_{i=1}^{n} f_i$ to parties $P_2, ..., P_n$, where $deg(p_1) = n \cdot \frac{k \cdot (k+1)}{2}$

Set intersection

5. All parties $P_1, ..., P_n$ perform Steps 1-3 of the protocol for the minimum of ranks composition scheme (see Figure 2) to calculate $S'_1 \cap ... \cap S'_n$ using polynomials
   $f'_i(X) = (X - r_{i1})^{n \cdot k} \cdot (X - r_{i2})^{n \cdot k} \cdot ... \cdot (X - r_{ik})^{n \cdot k}$ as input where $deg(f'_i(X)) = n \cdot k^2$
6. Party P_1 distributes the result $E(\lambda_n)$ with $p_2 = \lambda_n = \sum_{i=1}^{n} f'_i * (\sum_{j=0}^{c} q_{i+j,j})$ to $c+1$ randomly chosen parties $P_{j_1}, ..., P_{j_{c+1}}$, where $deg(q_{i+j,j}) = n \cdot k^2$ and $deg(p_2) = 2 \cdot n \cdot k^2$

Reduction step (t=nk-1,...,n-1)

7. Each party P_{j_s}, $(s = 1, ..., c+1)$
   a) calculates the encryption of $p_3 = p_1 * q''_{s,1} + p_2 * q''_{s,2}$ with random polynomials $q''_{s,1}, q''_{s,2}$ of degree $deg(p_2)$ such that $deg(p_3) = 4 \cdot n \cdot k^2$,
   b) performs Step 5 of the protocol for the minimum of ranks composition scheme on the encryption of polynomial p_3. The result is a reduced polynomial p^*_s.

Decryption

8. All parties perform a threshold decryption to obtain the polynomial $\Phi = \sum_{s=1}^{c+1} p^*_s = \sum_{l=0}^{t} p_3^{(l)} * F_l * (\sum_{s=1}^{c+1} q'_{s,l})$ where $deg(\Phi) = 4 \cdot n \cdot k^2 + t$.
9. Each party P_i $(i = 1, ..., n)$ determines the result of the function
   $Rd_t((S_1 \cup ... \cup S_n) \cap S'_1 \cap ... \cap S'_n)$ by checking for each $r_{il} \in S_i$ $(l = 1, ..., k)$ whether it appears in the polynomial Φ, meaning that the equation $(X - r_{il}) | \Phi$ holds. If no match is found, proceed with Step 7b) and a decreasing value of t.

Figure 3: Protocol for the sum of ranks composition scheme

does not lead to the desired type of protocol. This is due to the fact that after applying the union operation to the multisets, there may be elements in the union that are in fact not common to all parties. Using the additional intersection with $S'_1 \cap ... \cap S'_n$ eliminates this problem. The maximal multiplicity of a rule in the union $S_1 \cup ... \cup S_n$ is $n \cdot k$. In turn, $S'_1 \cap ... \cap S'_n$ contains all those elements with a multiplicity of $n \cdot k$ which all parties have in common. Thus, $(S_1 \cup ... \cup S_n) \cap S'_1 \cap ... \cap S'_n$ not only eliminates those elements that are not held by all parties but it also preserves the preferences with which the common input elements are held. Intuitively speaking, the protocol implementing Equation (7) works as follows:

1. Each party calculates polynomial representations of the multisets S_i and S'_i.
2. All parties calculate the set union on their input multisets $S_1, \ldots,\ S_n$. After completing this step, all parties hold an encrypted polynomial that represents $S_1 \cup ... \cup S_n$. This union encodes not only all the input elements that are held by the parties but it also represents the sum of the preferences for each element across all parties.
3. All parties calculate the set intersection operation on input sets $S'_1, \ldots, S'_n$ and the outcome

of the previous Step 2. This step is necessary in order to ensure that those elements are eliminated from $S_1 \cup ... \cup S_n$ which are held by some but not all parties.

4. All parties iteratively participate in the element reduction by t with $t = nk - 1, ..., n - 1$ on the result of the previous Step 3. As in the case of the multi-party, privacy-preserving protocol for the minimum of ranks composition scheme, the purpose of this step is to maximize the preference order. This is ensured by possibly repeating this step and the following step for decreasing t.

5. All parties participate in the threshold decryption of the result of Step 4 and check whether at least one of its input elements is a root of the polynomial computed in the previous step. If this is the case, the common elements with the maximal sum of ranks are found. If this is not the case (i.e., t was too large and the polynomial computed as part of Step 4 corresponds to the empty set), the parties repeat Step 4 with a value t decreased by one.

Figure 3 details our protocol for the sum of ranks composition scheme. The protocol works as follows. In Step 1, all parties construct the polynomials that represent the multisets S_i. In Step 2, party P_1 sends the encrypted polynomial δ_1 to party P_2. In Step 3, starting with party P_2 (up to P_n), each party P_i calculates a part of the set union using the received encryption of the polynomial δ_{i-1} and its own polynomial f_i. In Step 4, party P_1 receives and distributes the encrypted polynomial $p_1 = \delta_n$. Steps 1-4 correspond to the calculation of the function $S_1 \cup ... \cup S_n$.

In Step 5, the n parties perform the set intersection operation on the n sets $S'_1, ..., S'_n$ to obtain $S'_1 \cap ... \cap S'_n$. Party P_1 publishes the encryption of the polynomial $p_2 = \lambda_n$ to $c + 1$ randomly chosen parties (Step 6).

In Step 7, each of the chosen parties P_{j_s} $(s = 1, ..., c + 1)$ determines the polynomial $p_3 = p_1 * q''_{s,1} + p_2 * q''_{s,2}$ using the set intersection on two encrypted polynomials. Note that the degree of polynomial p_1 and p_2 differs. The random polynomials $q''_{s,1}, q''_{s,2}$ are therefore chosen of degree $deg(p_2)$ which is the larger degree of the polynomials p_1, p_2 since $deg(p_1) = n \cdot \frac{k\cdot(k+1)}{2} < 2n \cdot k^2 = deg(p_2)$ holds $\forall k \geq 1, n \geq 2$. Kissner and Song only proved the set intersection operation for two encrypted polynomials of the same degree [10]. Hence, we provide a formal proof of the correctness of the set intersection operation for two encrypted polynomials of different degree. A formal proof is given in Appendix A. We show that the set intersection operation $f * r + g * s$ for two encrypted polynomials f, g is still valid if we choose the random polynomials r, s of degree $max\{deg(f), deg(g)\}$. In our protocol this is exactly the chosen degree $deg(p_2)$ for the random polynomials $q''_{s,1}, q''_{s,2}$ as $max\{deg(p_1), deg(p_2)\} = deg(p_2)$, $\forall k \geq 1, n \geq 2$.

Next, the parties perform the reduction step by t on polynomial p_3. Specifically (see Section 3.4.1), the $c+1$ parties determine the $1, ..., t$-th derivatives to construct the polynomial p^*_s which is the reduction by t of polynomial p_3. The result p^*_s is broadcast to all parties. They then perform a threshold decryption in Step 8 to obtain polynomial Φ.

Finally, each party P_i checks whether the resulting polynomial Φ has at least one of its input elements r_{il} $(1 \leq l \leq k)$ as root. If this is the case, all these elements maximize the sum of ranks composition scheme. If this is not the case, the parties reduce the value t by one and repeat Steps 7b-9.

Correctness and Privacy. The above protocol combines the multiset union with the multiset intersection operation and multiple element reduction operations.

Steps 1-4 correspond to the union operation on n multisets which, by construction (see Section 3.4.1), returns a multiset containing each rule r exactly s times where s is the sum of the

Protocol	Communication	Computation	Party
$3PR^{MM}$ [13]	$O(1^l \cdot k^2)$	$O(1^l \cdot k^2)$	2-party
$3PR^{SR}$ [13]	$O(1^l \cdot k^2)$	$O(1^l \cdot k^2)$	2-party
MPROS^{MR}(Round-based)	$O(1^l \cdot c \cdot n \cdot k^n)$	$O(1^l \cdot c \cdot n \cdot k^n)$	multi-party
MPROS^{MR}(Multiset-based)	$O(1^l \cdot c \cdot n \cdot k^3)$	$O(1^l \cdot (c \cdot k^6 + n \cdot c \cdot k^4))$	multi-party
MPROS^{SR}(Round-based)	$O(1^l \cdot c \cdot n \cdot k^n)$	$O(1^l \cdot c \cdot n \cdot k^n)$	multi-party
MPROS^{SR}(Multiset-based)	$O(1^l \cdot c \cdot n^3 \cdot k^3)$	$O(1^l \cdot n^4 \cdot c \cdot k^6)$	multi-party

Table 1: Summary of protocol complexities

occurrences of r in each input set. As mentioned above, this multiset may contain rules that are not shared by all parties. Intersecting the result of the union operation with the intersection of the multisets S_i' eliminates these non-shared rules while preserving the (sum of the) preferences. To obtain the rules that maximize the sum of ranks of all parties, we apply the reduction step, this time starting with $t = nk - 1$ since the maximum possible sum of the assigned rank for a rule is nk. Assuming a rule r occurring d times in $(S_1 \cup ... \cup S_n) \cap S_1' \cap ... \cap S_n'$, the reduction by t (see Section 3.4.1) returns all elements that appear $max\{d - t, 0\}$ times. Due to the order in which the reduction is applied (i.e., for decreasing $t \geq n - 1$), the correctness of the protocol is guaranteed. In our protocol, each party P_i learns all maximally preferred common input elements and the value t corresponding to the reduction in which these rules were found. In order to prove that the new protocol is privacy-preserving in the semi-honest model, we can build on the results by Kissner et al. [10]. They show that in the presence of at most c colluding attackers, their multiset operations (union, intersection, and element reduction) are privacy-preserving in the semi-honest model and that these operations can be arbitrarily composed in a privacy-preserving manner. As our protocol combines the union and the intersection with several element reduction operations, it is privacy-preserving in the semi-honest model in the presence of at most c attackers as well. A performance analysis of the protocol can be found in Section 5 and Appendix 5.2.

5. Performance Comparison

In this section, we compare the performance of our new protocols for the round-based and the multiset-based approaches. The details of how we obtained these performance results are provided in Section 5.1 and 5.2. Table 1 summarizes the computational complexity in terms of encryption, decryption and homomorphic operations on ciphertexts and the communication overhead in the number of ciphertexts for the newly-developed protocols, including the two-party versions of our protocols. Recall that k denotes the number of input elements, n the number of parties, c the maximum number of colluding attackers and l the security parameter. All complexities are based on a worst-case analysis in the semi-honest model. Our newly-developed multiset-based protocols are polynomial-time bounded with respect to the number of parties and input elements, whereas the round-based constructions have exponential runtime with respect to the number of parties. Furthermore, the table shows that for more than three parties the communication overhead of the multiset-based protocols is smaller than the communication overhead of the round-based approach. For more than five parties the computation overhead of the multiset-based protocol for the minimum of ranks composition scheme is smaller than that of the round-based construction. For the sum of ranks composition scheme this holds for more

Step	Comm. complex. (M)	Type	Comp. complex.
1b)	$n \cdot c \cdot [\frac{k \cdot (k+1)}{2} + 1]$	EOC_p	$n \cdot [\frac{k \cdot (k+1)}{2} + 1]$
1d)	-	$+_{h_p}$	$n \cdot c \cdot [k \cdot (k+1) + 1]$
	-	$\times_{h_p}$	$n \cdot (c+1) \cdot [\frac{(k \cdot (k+1)+1) \cdot (k \cdot (k+1)+2)}{2}]$
	-	$+_{h_p}$	$n \cdot (c+1) \cdot [\frac{(k \cdot (k+1)) \cdot (k \cdot (k+1)+1)}{2}]$
2/3	$n \cdot [k \cdot (k+1) + 1]$	$+_h$	$(n-1) \cdot [k \cdot (k+1) + 1]$
4	$(c+1) \cdot [k \cdot (k+1) + 1]$	-	-
5	$\sum_{t=0}^{k-1} (c+1) \cdot (n-1) \cdot [k \cdot (k+1) + t + 1]$	$\times_{h_p}$	$(c+1) \cdot [(k-1) \cdot k \cdot (k+1) - \frac{(k-2) \cdot (k-1)}{2})$
		$+_{h_p}$	$\sum_{t=0}^{k-1} (c+1) \cdot [t \cdot k \cdot (k+1) + \frac{t \cdot (t+1)}{2}]$
		$\times_{h_p}$	$\sum_{t=0}^{k-1} (c+1) \cdot [\sum_{l=0}^{t} (4)] \rightarrow O(c \cdot k^6)$
			$deg(f_1) = d_1 = k \cdot (k+1) - l$, $deg(f_2) = d_2 = 2 \cdot l$
		$+_{h_p}$	$\sum_{t=0}^{k-1} (c+1) \cdot [\sum_{l=0}^{t} (5)] \rightarrow O(c \cdot k^6)$
			$deg(f_1) = d_1 = k \cdot (k+1) - l$, $deg(f_2) = d_2 = 2 \cdot l$
6	-	$+_h$	$\sum_{t=0}^{k-1} n \cdot c \cdot [k \cdot (k+1) + t + 1]$
	-	DEC	$\sum_{t=0}^{k-1} n \cdot [k \cdot (k+1) + t + 1]$
7	-	PE_p	$n \cdot k^2$
Overall	$O(n \cdot c \cdot k^3)$		$O(c \cdot k^6 + n \cdot c \cdot k^4)$
Overall$_p$	$-$		$O(k^6 + c \cdot k^4 + n \cdot c \cdot k^3)$

Table 2: Worst-case analysis of the protocol for the minimum of ranks composition scheme. The overall worst-case complexity corresponds to the executing of at most k rounds ($t = k-1, ..., 0$).

than six parties.

Obviously, our current multiset-based construction has limitations w.r.t. scalability for an increasing number of rules k each party holds. Especially for time-critical applications the practical benefit of our solution is expected to be limited. However, for less time-sensitive applications that allow an upper bound regarding the participating parties and number of rules (like in the Doodle case), our solutions are expected to perform reasonably well considering the fact that our solutions are polynomial-time bounded with respect to the number of parties n and the number of rules k. To the best of our knowledge, there are no other (more efficient) solutions for fair privacy-preserving reconciliation that consider the parties' preferences and allow for maximizing them.

5.1. Analysis for Minimum of Ranks

Next, we provide the details for our worst-case performance analysis of the two new multiset-based protocols. We determine the communication overhead by the number of ciphertexts exchanged between all parties. The size of the ciphertext depends on the used homomorphic cryptosystem and is not further specified here. To determine the computation overhead, we count encryption of polynomial coefficients (EOC), homomorphic add operations ($+_h$), homomorphic scalar operations ($\times_h$), decryption operations (DEC) and polynomial evaluations (PE). Some steps in our protocols allow several parties to compute operations in parallel. We indicate this potential speed up in the tables by an index p and provide the overall computational complexity both with and without parallel computation.

Table 2 shows the performance analysis of the multiset-based protocol for the minimum of ranks composition scheme. We first consider the communication overhead. In Step 1b) $n \cdot c$ messages are sent, as each party sends its encrypted polynomial to c parties. The degree of each of these polynomials is $\frac{k \cdot (k+1)}{2}$, so each message contains $(\frac{k \cdot (k+1)}{2} + 1)$ ciphertexts. In Steps 2 and 3 n messages are exchanged due to the ring-wise structure of the protocol. Each of these messages contains the encrypted coefficients of a polynomial of degree $k \cdot (k+1)$. In Step 4, party P_1 distributes $E(\lambda_n)$ to $c+1$ parties. Each of these parties calculates the reduction of the polynomial and sends it to all other parties resulting in $(c+1) \cdot (n-1)$ messages. The degree of the transmitted polynomials p_s^* is $k \cdot (k+1) + t$. This step will be repeated at most k times ($t = k-1, ..., 0$). Overall, the worst-case communication complexity is $O(n \cdot c \cdot k^3)$. Thus, considering the communication overhead the protocol is more efficient than the round-based approach for $n \geq 3$ parties.

Next, we consider the computational overhead. In Step 1b), each party computes the encryption of its polynomial. The degree of each polynomial is $\frac{k \cdot (k+1)}{2}$. This results in a total of $n \cdot [\frac{k \cdot (k+1)}{2} + 1]$ encryptions. In Step 1d), each party computes a part of the set intersection. The outer sums are computed over encrypted polynomials. Each sum requires in total $(max\{deg(f_1), deg(f_2)\} + 1)$ $+_h-$operations (see Section 3.4.2). Overall, with $c+1$ terms and n parties, we obtain $n \cdot c \cdot [k \cdot (k+1) + 1]$ $+_h$-operations. Next, the parties compute $c+1$ products of unencrypted and encrypted polynomials. Computing the products (see Section 3.4.2) requires $\times_h$ and $+_h$-operations. All operations in Step 1 can be calculated by the n parties in parallel. In Step 3, parties $P_2, ..., P_n$ first compute a sum of two encrypted polynomials of degree $k \cdot (k+1)$. In Step 5, $c+1$ parties compute the $1, ..., t$-th derivatives of the polynomial p of degree $k \cdot (k+1)$ using $\times_h$-operations. The polynomial p_s^* is calculated by the outer sum over t terms with the degrees of the polynomials varying from $k \cdot (k+1)$ to $k \cdot (k+1) + t$ and inner products of unencrypted and encrypted polynomials. Step 5 can be calculated in parallel by the $c+1$ parties. In Step 6, the polynomial Φ is calculated by a sum over the $c+1$ encrypted polynomials p_s^* with degree $(k \cdot (k+1) + t)$. In the worst-case, this operation is repeated k times. Φ is then decrypted using the threshold decryption. Since there are at most k rounds, these operations are also repeated at most k times. Finally, the polynomial Φ is evaluated at k points. Overall, the computation complexity is $O(c \cdot k^6 + n \cdot c \cdot k^4)$. With parallelism the overall computation cost is $O(k^5 + c \cdot k^4 + n \cdot c \cdot k^3)$.

5.2. Analysis for Sum of Ranks

Table 3 shows the performance analysis of the multiset-based protocol for the sum of ranks composition scheme.

We start with the communication overhead. In Step 2) party P_1 sends the encryption of δ_1 of degree $\frac{k(k+1)}{2}$ to party P_2 which results in $\frac{k(k+1)}{2} + 1$ ciphertexts. In Step 3), each party $P_i, i = 2, ..., n$ calculates the encryption of δ_i and sends the result to the next party P_{i+1}. Note that the degree of the polynomials sent increases by a value of $\frac{k \cdot (k+1)}{2}$ after each calculation of δ_i. Overall we obtain $\sum_{i=1}^{n-1}((i+1) \cdot \frac{k(k+1)}{2} + 1)$ ciphertexts. In Step 4, party P_1 distributes $E(p_1 = \delta_n)$ to all other parties where the degree of polynomial p_1 is $n \cdot \frac{k \cdot (k+1)}{2}$. In Step 5) $n \cdot c$ messages are sent, as each party sends its encrypted polynomial f_i' to c parties. The degree of each of these polynomials is $n \cdot k^2$, so each message contains $n \cdot k^2 + 1$ ciphertexts. Furthermore, n messages are exchanged due to the ring-wise structure of the set intersection step. Each of these messages contains the encrypted coefficients of a polynomial of degree $2n \cdot k^2$. In Step 6, party P_1 distributes $E(p_2 = \lambda_n)$ to $c+1$ parties where the degree of polynomial p_2 is $2n \cdot k^2$. Each of these parties calculates the reduction of the polynomial p_3 where $deg(p_3) = 4nk^2$ and sends it to all other parties resulting in $(c+1) \cdot (n-1)$ messages. The degree of the transmitted polynomials p_s^* is $4n \cdot k^2 + t$. This step will be repeated at most k times ($t = nk-1, ..., n-1$) (Step 7). Overall, the worst-case communication complexity is $O(n^3 \cdot c \cdot k^3)$.

Step	Comm. complex.	Type	Comp. complex.
1/2	$\frac{k\cdot(k+1)}{2}+1$	EOC	$\frac{k\cdot(k+1)}{2}+1$
3	$\sum_{i=2}^{n}(i\cdot\frac{k(k+1)}{2}+1)$	$\times_h$	$\sum_{l=1}^{n-1}(5)\rightarrow O(n^3\cdot k^4)$
			$deg(f_1)=\frac{k\cdot(k+1)}{2}\cdot l,\, deg(f_2)=\frac{k\cdot(k+1)}{2}$
		$+_h$	$\sum_{l=1}^{n-1}(4)\rightarrow O(n^3\cdot k^4)$
			$deg(f_1)=\frac{k\cdot(k+1)}{2}\cdot l,\, deg(f_2)=\frac{k\cdot(k+1)}{2}$
4	$(n-1)\cdot[n\cdot\frac{k\cdot(k+1)}{2}+1]$		-
5	$n\cdot c\cdot[n\cdot k^2+1]$	EOC	$n\cdot[n\cdot k^2+1]$
	$n\cdot[2\cdot n\cdot k^2+1]$	$+_{h_p}$	$n\cdot c\cdot[2\cdot n\cdot k^2+1]$
	-	$\times_{h_p}$	$n\cdot(c+1)\cdot[\frac{(n\cdot k^2+1)\cdot(n\cdot k^2+2)}{2}]$
	-	$+_{h_p}$	$n\cdot(c+1)\cdot[\frac{(n\cdot k^2)\cdot(n\cdot k^2+1)}{2}]$
		$+_{h_p}$	$(n-1)\cdot[2\cdot n\cdot k^2+1]$
6	$(c+1)\cdot[2\cdot n\cdot k^2]$		-
7	$\sum_{t=n-1}^{nk-1}(c+1)\cdot(n-1)\cdot[4\cdot n\cdot k^2+t+1]$	$+_h$	$(c+1)\cdot[4\cdot n\cdot k^2]$
		$\times_h$	$(c+1)\cdot[(5)+(5)']\rightarrow O(c\cdot n^2\cdot k^4)$
			$deg(f_1)=n\frac{k(k+1)}{2},\, deg(f_2)=2nk^2, deg(f_1')=2nk^2, deg(f_2')=2nk^2$
		$+_h$	$(c+1)\cdot[(4)+(4)']\rightarrow O(c\cdot n^2\cdot k^4)$
			$deg(f_1)=n\frac{k(k+1)}{2},\, deg(f_2)=2nk^2, deg(f_1')=2nk^2, deg(f_2')=2nk^2$
		$\times_h$	$(c+1)\cdot[(nk-1)\cdot 4\cdot n\cdot k^2-\frac{(nk-2)\cdot(nk-1)}{2}]$
		$+_h$	$\sum_{t=n-1}^{nk-1}(c+1)\cdot[t\cdot 4\cdot n\cdot k^2+\frac{t\cdot(t+1)}{2}]$
		$\times_h$	$\sum_{t=n-1}^{nk-1}(c+1)\cdot[\sum_{l=0}^{t}(5)]\rightarrow O(n^4\cdot c\cdot k^6)$
			$deg(f_1)=4n\cdot k^2-l,\, deg(f_2)=2\cdot l$
		$+_h$	$\sum_{t=n-1}^{nk-1}(c+1)\cdot[\sum_{l=0}^{t}(4)]\rightarrow O(n^4\cdot c\cdot k^6)$
			$deg(f_1)=4n\cdot k^2-l,\, deg(f_2)=2\cdot l$
8		$+_h$	$\sum_{t=n-1}^{nk-1}n\cdot c\cdot[4\cdot n\cdot k^2+t+1]$
		DEC	$\sum_{t=n-1}^{nk-1}n\cdot[4\cdot n\cdot k^2+t+1]$
9		PE	$(nk-(n-1))\cdot n\cdot k$
Overall	$O(n^3\cdot c\cdot k^3)$		$O(n^4\cdot c\cdot k^6)$
Overall$_p$	$-$		$O(n^4\cdot k^6+n^2\cdot k^4\cdot c)$

Table 3: Analysis of the protocol for the sum of ranks composition scheme. The overall worst-case complexity corresponds to the executing of at most $nk-(n-1)$ rounds ($t=nk-1,...,n-1$).

Next, we consider the computational overhead. In Step 1/2), party P_1 computes the encryption of its polynomial. The degree of the polynomial is $\frac{k\cdot(k+1)}{2}$. This results in a total of $\frac{k\cdot(k+1)}{2}+1$ encryptions. In Step 3), each party computes a part of the set union. Each party P_i computes δ_i by receiving the encrypted polynomial f_{i-1} and multiplying it with their own unencrypted polynomial f_i. Overall, we have $n-1$ products of unencrypted and encrypted polynomials where the degree of the polynomials differs for each party. Computing the products (see Section 3.4.2) requires $\times_h$ and $+_h$-operations. All operations can be calculated by the $n-1$ parties in parallel. In Step 5), each party computes the encryption of its polynomial f_i'. The degree of each polynomial is $n\cdot k^2$. This results in a total of $n\cdot[n\cdot k^2+1]$ encryptions. Furthermore, each party computes a part of the set intersection to calculate $S_1'\cap ... \cap S_n'$. The same argumentation holds as for the minimum of ranks composition scheme. Each sum requires in total $(max\{deg(f_1), deg(f_2)\}+1)$ $+_h-$operations (see Section 3.4.2). Overall, with $c+1$ terms and n parties, we obtain $n\cdot c\cdot[2nk^2+1]$ $+_h$-operations. Next, the parties compute $c+1$ products of unencrypted and encrypted polynomials. Computing the products (see Section 3.4.2) requires $\times_h$ and $+_h$-operations. Finally, parties $P_2, ..., P_n$ compute a sum of two encrypted polynomials of degree $2nk^2$. All operations in Step 5 can be calculated by the n parties in parallel. In Step 7a), c+1 parties calculate the encryption of polynomial p_3 where $deg(p_3) = 4nk^2$ since the degrees of polynomials $p_1, p_2, q''_{s,1}, q''_{s,2}$ are given by $deg(p_1) = n\frac{k(k+1)}{2}, deg(p_2) = deg(q''_{s,1}) = deg(q''_{s,2}) = 2nk^2$. So, for the outer sum we need $4nk^2$ $+_h$-operations calculated by $c+1$ parties. Similar to Step 3, the multiplication of encrypted and unencrypted polynomials needs $\times_h$ and $+_h$-operations. Next, the $c+1$ parties compute the $1, ..., t$-th derivatives of the polynomial p_3 of degree $4nk^2$ using $\times_h$-operations. The polynomial p^*_s is calculated by the outer sum over t terms with the degrees of the polynomials varying from $4nk^2$ to $4nk^2 + t$ and inner products of unencrypted and encrypted polynomials. Step 7 can be calculated in parallel by the $c+1$ parties. In Step 8, the polynomial Φ is calculated by a sum over the $c+1$ encrypted polynomials p^*_s with degree $(4n \cdot k^2 + t)$. In the worst-case, this operation is repeated $nk - (n-1)$ times. Φ is then decrypted using the threshold decryption. Since there are at most $nk - (n-1)$ rounds, these operations are also repeated at most $nk - (n-1)$ times. Finally, the polynomial Φ is evaluated at k points. Overall, the computation complexity is $O(n^4 \cdot c \cdot k^6)$. With parallelism the overall computation cost is $O(n^4 \cdot k^6 + n^2 \cdot k^4 \cdot c)$.

6. Conclusions and Future Work

In this paper we have proposed four new protocols for privacy-preserving, fair reconciliation of ordered sets for multiple parties. As a next step, we plan to implement and test the performance of our new protocols. In this context, we plan to compare different homomorphic cryptosystems w.r.t. their efficiency. In addition, we will explore some potential optimizations such as finding a way to pre-compute the threshold value t in order to eliminate the iterative reduction steps in the multiset-based constructions. This may lead to a considerable performance improvement of our protocols. Furthermore, we will investigate whether it is possible to modify the multiset-based protocols such that the protocol output is limited to one and not all possible results. Also, a more general solution where party P_i holds k_i elements which may differ would be nice. Finally, two important directions for our future research are the design of protocol variants that are secure in the malicious model and the investigation of different notion of fairness like pareto-optimal solutions.

Acknowledgments

In part, this work was supported by NSF Award CCF 1018616, DFG and the UMIC Research Center, RWTH Aachen.

References

[1] J. Camenisch and G. M. Zaverucha. Private Intersection of Certified Sets. In *Financial Cryptography*, pages 108–127, 2009.

[2] R. Cramer, I. Damgard, and J. B. Nielsen. Multiparty Computation from Threshold Homomorphic Encryption. In *Proceedings of EUROCRYPT'01*, pages 280–299. Springer-Verlag, 2001.

[3] E. Cristofaro and G. Tsudik. Practical Private Set Intersection Protocols with Linear Computational and Bandwidth Complexity. In *Financial Cryptography and Data Security 2010*, 2010.

[4] Doodle Easy Scheduling. http://www.doodle.com/.

[5] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In *Proceedings of EUROCRYPT'04*, 2004.

[6] O. Goldreich, S. Micali, and A. Wigderson. How to Play ANY Mental Game. In *Proceedings of STOC '87 ACM Conference on Theory of Computing.* ACM, 1987.

[7] C. Hazay and Y. Lindell. Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. Cryptology ePrint Archive, Report 2009/045, 2009. http://eprint.iacr.org/.

[8] S. Hohenberger and S. A. Weis. Honest-Verifier Private Disjointness Testing Without Random Oracles. In *Privacy Enhancing Technologies*, pages 277–294, 2006.

[9] S. Jarecki and X. Liu. Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In *TCC*, pages 577–594, 2009.

[10] L. Kissner and D. X. Song. Privacy-Preserving Set Operations (Last modified June 2006). In *CRYPTO*, pages 241–257, 2005.

[11] R. Li and C. Wu. An Unconditionally Secure Protocol for Multi-Party Set Intersection. In *Proceedings of ACNS '07 Applied Cryptography and Network Security*, pages 226–236, Berlin, Heidelberg, 2007. Springer-Verlag.

[12] U. Meyer, S. Wetzel, and S. Ioannidis. Distributed Privacy-Preserving Policy Reconciliation. In *ICC*, pages 1342–1349, 2007.

[13] U. Meyer, S. Wetzel, and S. Ioannidis. New Advances on Privacy-Preserving Policy Reconciliation. In *iacr eprint 2010/64*, 2010. http://eprint.iacr.org/2010/064.

[14] G. S. Narayanan, T. Aishwarya, A. Agrawal, A. Patra, A. Choudhary, and C. P. Rangan. Multi Party Distributed Private Matching, Set Disjointness and Cardinality of Set Intersection with Information Theoretic Security. In *Cryptology and Network Security*, pages 21–40, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] G. Neugebauer, U. Meyer, S. Wetzel. Fair and Privacy-Preserving Multi-Party Protocols for Reconciling Ordered Input Sets. In 13th Information Security Conference (ISC), 2010.

[16] A. Patra, A. Choudhary, , and C. P. Rangan. Round Efficient Unconditionally Secure MPC and Multiparty Set Intersection with Optimal Resilience. In *Progress in Cryptology - INDOCRYPT 2009*, pages 398–417, 2009. Springer-Verlag.

[17] Yingpeng Sang and Hong Shen. Efficient and secure protocols for privacy-preserving set operations. In ACM Trans. Information and System Security Volume 13, Issue 1. 2009.

[18] A. Shamir. How to Share a Secret. In *Communications of the ACM*, volume 22, pages 612–613. ACM, 1979.

A. Proofs

In the sum of ranks protocol, we need a generalization of Kissner's Lemma 2 (page 26-27) [10] that holds for two polynomials of different degree. This generalized Lemma is proven in the following.

Lemma A.1 *Let f, g be polynomials in $R[x]$ where R is a ring such that no PPT adversary can find the size of its subfields with non-negligible probability, $deg(f) = \alpha, deg(g) = \gamma, \beta \geq \alpha \geq \gamma, gcd(f, g) = 1$, and $f[deg(f)] \in R^* \wedge g[deg(g)] \in R^*$. Let $r = \sum_{i=0}^{\beta} r[i]x^i$ and $s = \sum_{i=0}^{\beta} s[i]x^i$, where $\forall_{0 \leq i \leq \beta}\ r[i] \leftarrow R$, $\forall_{0 \leq i \leq \beta}\ s[i] \leftarrow R$ (independently).*

Let $u = f * r + g * s = \sum_{i=0}^{\alpha+\beta} u[i]x^i$. Then $\forall_{0 \leq i \leq \alpha+\beta}\ u[i]$ are distributed uniformly and independently over R.

Proof. For clarity, we briefly outline the idea of the proof. Our goal is to calculate the number z of r, s pairs such that $f*r+g*s = u$ for any fixed polynomials f, g, u with $gcd(f, g) = 1$. If the number of possible result polynomials u is equal to the total number of possible r, s pairs divided by z, then this implies that the coefficients of the result polynomial u are distributed uniformly if we choose the coefficients of r, s uniformly and independently from R. Let us assume there exists at least one pair $\hat{r}, \hat{s}$ for a specific u such that $f * \hat{r} + g * \hat{s} = u$. For any pair $\hat{r}', \hat{s}'$ such that $f * \hat{r}' + g * \hat{s}' = u$, it holds that

$$f * \hat{r} + g * \hat{s} = f * \hat{r}' + g * \hat{s}'$$
$$f * (\hat{r} - \hat{r}') = g * (\hat{s}' - \hat{s}).$$

As $gcd(f, g) = 1$, we conclude that $g|\hat{r} - \hat{r}'$ and $f|\hat{s}' - \hat{s}$ using Lemma 22 [10]. We may apply Lemma 22, since it is proven for polynomials of arbitrary degree. Let

$$p * g = \hat{r} - \hat{r}' \quad \wedge \quad p * f = \hat{s}' - \hat{s}. \tag{8}$$

On the one hand we have to show that there exists no pairs $\hat{r}', \hat{s}'$ such that $f*\hat{r}'+g*\hat{s}' = u$ that are not generated by a single choice of the polynomial p of degree at most $\beta - \alpha$. On the other hand we need to show that each polynomial p, of degree at most $\beta - \alpha$, determines exactly one unique pair $\hat{r}', \hat{s}'$ such that $f * \hat{r}' + g * \hat{s}' = u$.

We first show that the two equations in (8) can only be valid if the degree of p is at most $\beta - \alpha$. The degree of $\hat{r} - \hat{r}'$ is β and the degree of g is γ. The product of g and p yields a polynomial of degree $\gamma + deg(p)$. Since the result $\hat{r} - \hat{r}'$ is of degree β, $deg(p)$ can be $\beta - \gamma$ at most. The degree of $\hat{s}' - \hat{s}$ is β and the degree of f is α. The product of f and p yields a polynomial of degree $\alpha + deg(p)$. Since the result $\hat{s}' - \hat{s}$ is of degree β, $deg(p)$ can at most be $\beta - \alpha$. Since it holds that $\alpha \geq \gamma$, the two equations are only valid if $deg(p)$ is at most $\beta - \alpha$.

Now we show that there exists no pairs $\hat{r}', \hat{s}'$ such that $f*\hat{r}'+g*\hat{s}' = u$, that are not generated by some choice of one polynomial p of degree at most $\beta - \alpha$. Let $p' * g = \hat{r} - \hat{r}'$ and $p * f = \hat{s}' - \hat{s}$ be valid for any p', p. As we proved that $g|\hat{r} - \hat{r}'$ and $f|\hat{s}' - \hat{s}$, we can represent f and g as

$$f * (\hat{r} - \hat{r}') = g * (\hat{s}' - \hat{s})$$
$$f * (p' * g) = g * (p * f).$$

We apply Lemma 21 [10] as the leading coefficients of f and g are members of R^*.

$$f * (p' * g) = (g * p) * f$$
$$\Rightarrow p' * g = g * p \quad \text{(Lemma 21)}$$
$$\Rightarrow p' = p \quad \text{(Lemma 21)}$$

Thus we have shown our assumption since it holds that $p = p'$.

Finally we show that each polynomial p exactly determines one unique pair $\hat{r}', \hat{s}'$ such that $f*\hat{r}'+g*\hat{s}' = u$. It holds that $\hat{r}' = \hat{r}-g*p$, $\hat{s}' = \hat{s}+f*p$ and $f, g, \hat{r}, \hat{s}$ are fixed. Thus a choice of p determines both $\hat{r}', \hat{s}'$. The uniqueness is guaranteed due to Lemma 21 [10]. If these assignments were not unique, there would exist polynomials p, p' such that either $\hat{r}' = \hat{r} - g * p = \hat{r} - g * p'$ or $\hat{s}' = \hat{s} + f * p = \hat{s} + f * p'$ for some polynomials $p \neq p'$.

As a result the number of polynomials p, of degree at most $\beta - \alpha$, is exactly equivalent to the number of r, s pairs such that $f * r + g * s = u$ and there are $|R|^{\beta-\alpha+1}$ such polynomials p. There are $|R|^{2\beta+2}$ r, s pairs. As $\frac{|R|^{2\beta+2}}{z} = \frac{|R|^{2\beta+2}}{|R|^{\beta-\alpha+1}} = |R|^{\alpha+\beta+1}$ which is equivalent to the $|R|^{\alpha+\beta+1}$ possible result polynomials u. □

Thus we have proven that all $u[i]$ are distributed uniformly and independently over R. Note that the dominating polynomial f with the possibly higher degree yields the same condition for the polynomial p, which is "degree at most $\beta - \alpha$", as in Kissner's proof [10]. Also we choose both random polynomials r, s of the degree $\beta = max\{deg(f), deg(g)\}$ which yields the same r, s pair domain which is $|R|^{2\beta+2}$. The number of possible r, s pairs divided by z is $|R|^{\alpha+\beta+1}$ which is the same as the possible result polynomials u. The last argument holds since the degree of u is also dominated by the product of $f * r$.