Toward Optimal Player Weights in Secure Distributed Protocols*

K. Srinathan, C. Pandu Rangan, and V. Kamakoti

Department of Computer Science and Engineering, Indian Institute of Technology, Madras, Chennai - 600036, India,
ksrinath@cs.iitm.ernet.in, {rangan,kama}@iitm.ernet.in

Abstract. A secure threshold protocol for n players tolerating an adversary structure $\mathcal{A}$ is feasible iff $\max_{a\in\mathcal{A}}|a| < \frac{n}{c}$, where $c = 2$ or $c = 3$ depending on the adversary being eavesdropping (passive) or Byzantine (active) respectively [1]. However, there are situations where the threshold protocol Π for n players tolerating an adversary structure $\mathcal{A}$ may not be feasible but by letting each player P_i to act for a number of similar players, say w_i, a new secure threshold protocol Π' tolerating $\mathcal{A}$ may be devised. Note that the new protocol Π' has $N = \sum_{i=1}^{n} w_i$ players and works with the same adversary structure $\mathcal{A}$ used in Π. The integer quantities w_i's are called weights and we are interested in computing w_i's so that

1. Π' tolerates $\mathcal{A}$ even if Π does not tolerate $\mathcal{A}$.
2. $N = \sum_{i=1}^{n} w_i$ is minimum.

Since the best known secure threshold protocol over N players has a communication complexity of $\mathcal{O}(mN^2 \lg|\mathbf{F}|)$ bits [9], where m is the number of multiplication gates in the arithmetic circuit, over the finite field $\mathbf{F}$, that describes the functionality of the protocol, it is evident that the weights assigned to the players have a direct influence on the complexity of the resulting secure weighted threshold protocol. In this work, we focus on computing the optimum N. We show that computing the optimum N is NP-Hard. Furthermore, we prove that the above problem of computing the optimum N is inapproximable within $(1-\epsilon)\ln\left(\frac{|\mathcal{A}|}{c}\right) + \frac{\ln\left(\left(\frac{|\mathcal{A}|}{c}\right)^{(1-\epsilon)}\right)-1}{N^*}(c-1)$, for any $\epsilon > 0$ (and hence inapproximable within $\Omega(\lg|\mathcal{A}|)$), unless $NP \subset DTIME(n^{\log\log n})$, where N^* is the optimum solution.

1 Motivating Example

Consider a set of five players $\mathcal{P} = \{P_1, P_2, P_3, P_4, P_5\}$ involved in a secure distributed protocol wanting to tolerate the (passive) adversary structure $\mathcal{A}$ given by

$$\mathcal{A} = \{(1,2,3),(1,2,4),(1,5),(2,5),(3,4)\}$$

* The first author would like to thank Infosys Technologies Ltd., India for financial support.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 232–241, 2001.

From the results of [1], it is clear that $\mathcal{A}$ cannot be tolerated by any threshold protocol. Nevertheless, the above adversary can be tolerated by a threshold-type protocol among nine players where players P_1, P_2, P_3, P_4 and P_5 act for one, one, two, two and three players respectively. This is indeed so because the corruption of any one set in the adversary structure leads to the corruption of at the most four out of the nine players which is tolerable[1]. In this example, $n = 5, w_1 = w_2 = 1, w_3 = w_4 = 2, w_5 = 3, c = 2, N = \sum_{i=1}^{n} w_i = 9$.

2 Basic Definitions and Model

2.1 Secure Multiparty Computation

Consider a fully connected synchronous network of n players (processors), $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, who do not trust each other. Nevertheless they want to compute some agreed function of their inputs in a secure way. Security here means maintaining correctness of the output while keeping the players' inputs as private as possible, even if some of the players are faulty. This task can be easily accomplished if there exists a trusted third party. But assuming the existence of such a trusted third party is quite unrealistic. The goal of *secure multiparty computation* is to transform a given protocol involving a trusted third party into a protocol without need for the trusted third party, by simulating the trusted third party among the n players.

The players' distrust in each other and in the underlying network is usually modeled via an *adversary* that has control over some of the players and communication channels. Many different adversary models have been considered, each modeling different problems, or addressing a different setting. These approaches can be classified according to a number of criteria that are briefly discussed below. Adversaries are classified according to their *computational resources* (limited (cryptographic) or unli mited (information theoretic)), their control over communication (secure, insecure, or unauthenticated channels), their control over corrupted players (eavesdropping (passive), fail-stop, or Byzantine (active)), their mobility (static, adaptive, or mobile) and their corruption capacity (threshold or non-threshold). In the information theoretic model one can distinguish between protocols with small (*unconditional*) or zero (*perfect*) failure probability.

In the information theoretic setting, [1] gave a perfect protocol for the general secure multiparty computation problem in the synchronous secure channels model without broadcast and proved tight bounds on the number of corrupted players that can be tolerated.

Theorem 1 ([1]). *For every $n \geq 2$, there exist Boolean functions f such that there is no synchronous $\lceil \frac{n}{2} \rceil$-secure protocol for n players that computes f. For every $n \geq 3$, there exist functions f such that no synchronous protocol for n players $\lceil \frac{n}{3} \rceil$-securely computes f, if Byzantine adversaries are allowed.* ∎

2.2 The Adversary Model

In this section, we formally define the *weighted threshold adversaries*. We begin with a brief look at the *threshold* adversaries.

Threshold Adversaries. A threshold adversary, $\mathcal{A}$, is a probabilistic strategy, that can *corrupt* up to $t < n$ among the n players involved in the protocol. The *corruption* may be either *active* or *passive*, by which we mean the following:

1. *Passive Corruption:* The adversary in this case behaves like an *eavesdropper*; that is, the adversary can gather all the information present with the corrupted players and can also perform any arbitrary computation on these gathered data.
2. *Active Corruption:* The adversary here is also referred to as a *Byzantine* adversary. They can do all what an eavesdropping adversary can and in addition can also take complete control of the corrupted players and alter the behaviour of the corrupted players in an arbitrary and coordinated fashion.

Tolerable Threshold Adversaries: It is known that all the passive threshold adversaries such that $t \leq \left\lfloor \frac{n-1}{2} \right\rfloor$ can be tolerated. That is, it is possible to construct multiparty computation protocols that are *secure* against such an adversary. By a security against an adversary $\mathcal{A}$, we mean, whatever $\mathcal{A}$ does in the protocol, the same effect (on the output) could be achieved by an adversary (may be different from $\mathcal{A}$ but similar to it in costs) in the ideal protocol (that assumes the existence of a trusted third party to whom all the inputs can be sent and outputs received). For more formal and "correct" definitions of security, we refer the readers to [2,6,10]. Similarly, in the case of active adversaries, we require that $t \leq \left\lfloor \frac{n-1}{3} \right\rfloor$.

Generalized Adversaries. In contrast to the threshold adversaries, [7,8] introduced a more general adversary characterized by a monotone adversary structure which is a set of subsets of the player set, wherein the adversary may corrupt the players of one set in the structure. An adversary structure is said to satisfy the $\mathcal{Q}^{(c)}$ property if no c sets in the adversary structure cover the full set of players. It is proved that in the passive model, every function can be computed securely with respect to a given adversary structure if and only if the adversary structure satisfies the $\mathcal{Q}^{(2)}$ property. Similarly, in the active model, a secure computation of a function is possible if and only if the adversary structure satisfies the $\mathcal{Q}^{(3)}$ property.

Weighted Threshold Adversaries. The weighted threshold adversaries are somewhere in between the threshold and the generalized adversaries. These adversaries are characterized by adversary structures that possess the following addition property so that they are tolerable: for each player P_i, $1 \leq i \leq n$, there exists a non-negative weight w_i, such that the adversary structure is tolerated in a threshold-type protocol with $N = \sum_{i:P_i \in \mathcal{P}} w_i$ players. Hereafter in the sequel,

unless explicitly specified, we will use the term adversary structure to mean the maximal basis[1].

In the weighted threshold adversary setting, one of the ways to improve the complexity of the resulting secure protocol is to assign weights to each of the players so that the adversary can be tolerated with the sum of the weights kept at a minimum, since, a larger sum of weights calls for larger number of secret shares (essentially of the same size) and hence an increase in the computation and communication complexities.

3 The Optimal Player Weights Problem

In this section, we define the problem of assigning optimum weights to the players in a secure multiparty protocol tolerating weighted threshold adversaries.

Definition 1 (Optimum Assignment of Player Weights(OAPW)). *Given the player set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, the adversary structure $\mathcal{A} \subset 2^{\mathcal{P}}$, and a constant c,[2] a valid assignment of player weights (if it exists) is a function $f : \mathcal{P} \to Z^+ \cup \{0\}$ such that for all sets $z \in \mathcal{A}$, $\sum_{P_i \in z} f(P_i) < \frac{\sum_{\forall P_i \in \mathcal{P}} f(P_i)}{c}$. A valid assignment of player weights, f, is said to be an optimum assignment of player weights if there does not exist any valid assignment of player weights, f', such that $\sum_{\forall P_i \in \mathcal{P}} f'(P_i) < \sum_{\forall P_i \in \mathcal{P}} f(P_i)$.*

Definition 2 (Decision Version of the OAPW problem).

INSTANCE: *A finite set $\mathcal{P}$, a collection $\mathcal{A}$ of subsets of $\mathcal{P}$, a constant c, and a positive integer k.*
QUESTION: *Does there exist a valid assignment of player weights $f : \mathcal{P} \to Z^+ \cup \{0\}$ such that $\sum_{\forall P_i \in \mathcal{P}} f(P_i) > ck$ and for all sets $z \in \mathcal{A}$, $\sum_{P_i \in z} f(P_i) \leq k$?*

From the (refer Definition 2) constraint that for all sets $z \in \mathcal{A}$, $\sum_{\forall P_i \in z} f(P_i) \leq k$, it is obvious that we can restrict the range of the function f to $\{0, ..., k\}$. We denote an instance to the OAPW problem by the ordered list $< \mathcal{P}, \mathcal{A}, c, k, r >$, where $f : \mathcal{P} \to \{0, ..., r\}$, and the *size of the solution* to the above instance is $ck + 1$.

4 Hardness of the OAPW Problem

Definition 3 (Density Index Number of a Graph G). *: Given a simple undirected graph $G = (V, E)$, the c-Density Index Number of G is defined as*

$$c\text{-}DIN(G) = min \left\{ ck{+}1 \;\middle|\; \begin{array}{l} \text{there exists } V' \subseteq V, |V'| = ck{+}1 \text{ such that there does } \textbf{not} \\ \text{exist a vertex } u \in V \text{ adjacent to } \geq k{+}1 \text{ vertices in } V'. \end{array} \right\}$$

[1] Given the adversary structure $\mathcal{A}$, the maximal basis $\mathcal{A}_{basis} = \{z \in \mathcal{A} | \nexists z' \supset z, z' \in \mathcal{A}\}$.
[2] Note that $c = 2$ if the adversary is passive and $c = 3$ if the adversary is active.

Theorem 2. *Given a simple undirected graph $G = (V, E)$, the size of the minimum dominating set (*MDS*) of G is equal to* 1-DIN(G^c)*, where G^c is the complement of G.*

Proof: Let 1-DIN$(G^c) = k+1$ and $V' \subseteq V$ be the subset satisfying the property as defined in Definition 3 with $|V'| = k+1$. From definition of 1-DIN we see that every vertex in V (and hence in $V - V'$) in G^c is *not* adjacent to at least one vertex in V'. This implies that in G, every vertex in $V - V'$ is adjacent to at least one vertex in V'. Hence, V' is a dominating set of G. If V' is not the minimum dominating set, then let U be the minimum dominating set of G. Then, $|U| \leq k$ and any vertex $u \in V - U$ is adjacent to at least one vertex in U. Therefore in G^c, any vertex $u \in V - U$ is not adjacent to all the vertices in U. Also, since no vertex in U can be adjacent to all the vertices in U (due to the fact that a vertex cannot be adjacent to itself), the 1-DIN(G^c)=k which is a contradiction. Thus, the *minimum* constraint in the definition of 1-DIN implies that V' is indeed a MDS of G. ∎

The fact that computing the size of a MDS of a graph is NP-complete [5] and Theorem 2 imply the following theorem.

Theorem 3. *Given a simple undirected graph $G = (V, E)$, computing* 1-DIN(G) *is NP-complete.*

Theorem 4. *c-*DIN(G) *for any fixed constant c, where $G = (V, E)$ is a simple undirected graph, is NP-Hard.*

Proof: We reduce the problem 1-DIN(G) to the problem c-DIN(G). Given an instance of the problem 1-DIN(G), construct $G' = (V', E')$ containing c copies of $G = (V, E)$; $V' = V_1 \cup \cdots \cup V_c$ such that $i \in V_j$ is relabelled $< i, j >$. Similarly, E' containing c copies of E; $E' = E_1 \cup \cdots \cup E_c$ such that for every pair of vertices $(i, k) \in E_j$, is relabelled $(< i, j >, < k, j >)$. Solve the c-DIN problem on G'. By the Pigeonhole principle we see that, there exists a solution to $c\text{–}DIN(G')$ if and only if there exists a solution to the problem $1\text{–}DIN(G)$. ∎

Theorem 5. *Problem OAPW is NP-Hard.*

Proof: Given a simple undirected graph $G = (V, E)$, we suggest the following method for computing c-DIN(G). Without loss of generality let us assume that the vertices of G are numbered $\{ 1, \ldots, n \}$, $|V| = n$. Let the set $\mathcal{P} = V$. Construct the set $\mathcal{A} = \{V_1, \ldots, V_n\}$, such that, V_i is a set containing all vertices adjacent to vertex i in G. Solve the problem OAPW with the sets $\mathcal{P}$ and $\mathcal{A}$ as defined above. We now show that the function f does not exist for $ck+1 <$ c-DIN(G)$-c$. Let us assume that f exists and let $V' \subseteq P$ be the set of vertices that are assigned non-zero values by f.

Case 1: ($|V'| \leq k+1$). From Definition 3 (of c-DIN), there exist an $a \in A$ such that $V' \subseteq a$. This implies that $\sum_{i \in a} f(i) > k$, a contradiction.

Case 2: ($|V'| > k+1$). Consider a $V" \subseteq V'$, $|V"| = k+1$. From Definition 3 (of c-DIN), there exist an $a \in A$ such that $V'' \subseteq a$. This implies that $\sum_{i \in a} f(i) > k$, a contradiction.

For $ck + 1 =$ c-DIN(G)$-c$, consider any $V' \subseteq P$, $|V'| = ck + 1$. Assign $f(i) \leftarrow 1$, $\forall i \in V'$ and $f(i) \leftarrow 0$, $\forall i \notin V'$. It is easy to see that f is a solution to the $OAPW$ problem. The same concept can be extended to show that for $ck+1 >$ c-DIN(G)$-c$ there exist a function f satisfying the constraints specified by the $OAPW$ problem. Hence, given an algorithm for the $OAPW$ problem that takes $O(g(n))$ time, one can compute the c-DIN of a simple undirected graph G in $O(g(n) \log n)$ time. Since c-DIN(G) $\leq n$ it is enough to consider $ck + 1 \leq n$. The above discussion and Theorem 4 imply the proof of this theorem. ∎

5 An Approximate Algorithm for the OAPW Problem

In this section, we first reduce the $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ to $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$. We then provide an (exponential) algorithm to solve the $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ exactly followed by a (polynomial) approximate algorithm for the same. Finally, we design an approximate algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ and analyse its performance.

Theorem 6. *A solution to OAPW in which $f : \mathcal{P} \rightarrow \{0, 1\}$ implies a solution to OAPW in which $f : \mathcal{P} \rightarrow \{0, ..., k\}$.*

Proof: Given an instance $I = (\mathcal{P}, \mathcal{A}, c, k, k)$ of OAPW, construct an instance $I' = (\mathcal{P}', \mathcal{A}, c, k, 1)$ in which $\mathcal{P}'$ is k copies[3] of $\mathcal{P}$. Solving OAPW on I', each $P_i \in \mathcal{P}$ would have been assigned *at most* k $1s$ (at most once in each copy of $\mathcal{P}$). Computing $f(i)$ to be the number of $1s$ assigned to P_i in $\mathcal{P}'$, gives a solution for the OAPW on I. ∎

5.1 Solving $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$

Exact Solution Given the instance $I = (\mathcal{P}, \mathcal{A}, c, k, 1)$ of the OAPW problem, construct a bipartite graph $G = (X, Y, E_g)$ with $X = \mathcal{P}$, $Y = \mathcal{A}$. Add (x, y) to E_g if and only if $x \in X$, $y \in Y$, and $x \in y$ (that is, the player x is present in the set y). Now, the problem of $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ stated graph theoretically is to find $ck + 1$ vertices in X such that the degree of each vertex $y \in Y$ in the subgraph induced by these $ck + 1$ vertices union Y on G is $\leq k$.

Consider the bipartite graph $H = (X, Y, E_h)$ where $E_h = \{(x, y) | x \in X, y \in Y, (x, y) \notin E_g\}$. The $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ problem can now be rephrased as to find $ck + 1$ vertices in X such that the degree of each vertex $y \in Y$ in the subgraph induced by these $ck + 1$ vertices union Y on H is at least $(c-1)k + 1$.

From the bipartite graph $H = (X, Y, E_h)$, we construct the following instance of a set multi-cover problem that solves the $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ problem: Let

[3] Since the search space of k is bounded by a polynomial in $(|\mathcal{P}| + |\mathcal{A}|)$, the given construction is feasible.

Algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$

1. Given the instance $I_{oapw} = (\mathcal{P}, \mathcal{A}, c, k, 1)$ of the OAPW problem, construct the instance $I_{smc} = (U, \mathcal{F}, (c-1)k+1, ck+1)$ of the set multi-cover problem as illustrated in Subsection 5.1.
2. Solve the instance I_{smc} using the approximate set multi-cover algorithm of [3], which is a natural extension to the greedy approximate algorithm for the set cover problem.
3. The solution to the instance I_{smc} (if it exists) gives rise to the set of vertices in X (i.e. the set of players in $\mathcal{P}$) that are to be given the weight 1. The rest of the players are given the weight 0.
4. If the instance I_{smc} has no solution then the instance I_{oapw} has no solution as well.

Fig. 1. The Approximate Algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$

the set $U = Y$ and the family of subsets of U be $\mathcal{F} = \{X_i | i = 1, 2, \ldots, |X|\}$, where X_i denotes the set of all elements in Y that are adjacent to the i^{th} element in X in the bipartite graph H. The decision version of the $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ problem now reads as follows: *Does there exist* $ck+1$ *or less number of sets from* $\mathcal{F}$ *such that their union covers each element of* U *at least* $(c-1)k+1$ *times?*

The above problem can be solved using the solution to the set multi-cover problem which is as follows.

Definition 4 (Set Multi-Cover Problem).
INSTANCE: *A set* U*, a family* $\mathcal{F}$ *of set of subsets of* U*, positive integers* m *and* k*.*
QUESTION: *Does there exist* $\leq k$ *sets from* $\mathcal{F}$ *such that they together cover each element of* U *at least* m *times?*

Thus, based on the (exponential) algorithm of finding the minimum set multi-cover, we now have an (exponential) algorithm to solve the $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ problem.

Approximating $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$**.** We proceed by "replacing" the exponential algorithm of finding the minimum set multi-cover by its corresponding approximate greedy algorithm as proposed by [3]. Thus, the resulting approximate algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ is as given in Fig. 1.

Theorem 7. *The algorithm presented in Fig. 1 runs in time polynomial in the size of the input and correctly solves the* $OAPW(\mathcal{P}, \mathcal{A}, c, k, 1)$ *problem.*

Proof: Since each of the four steps in the algorithm runs in time polynomial in $(|\mathcal{P}| + |\mathcal{A}|)$, it is evident that the overall algorithm runs in time polynomial in the input size.

From the construction of Subsection 5.1, it is clear that a solution to the instance I_{oapw} exists if and only if the instance I_{smc} has a solution. We now

show that every solution to the instance I_{smc} leads to a solution to the instance I_{oapw}, thereby proving the theorem.

Let $(X_{i_1}, X_{i_2}, \ldots, X_{i_{ck+1}})$, $X_j \in \mathcal{F}$ be a set multi-cover of U such that their union covers U at least $(c-1)k+1$ times. We stress that the value of k here may be much larger than what the minimum set multi-cover requires it to be. From this (approximate) set multi-cover, we obtain the corresponding vertices in X that along with the vertices in Y induce a subgraph H_{sub} on H such that each vertex in Y in H_{sub} has a degree of at least $(c-1)k+1$. Therefore, since E_h and E_g compliment each other, there exist $ck+1$ vertices in X such that the degree of every vertex in Y is bounded by $\leq k$ in the subgraph G_{sub} induced by the $ck+1$ vertices of X along with Y on G: thus providing a solution to the instance I_{oapw}. ∎

Corollary 1. *The approximate algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ (see Fig. 2) follows from the Theorems 6 and 7.*

6 Inapproximability Results regarding the OAPW Problem

We begin with the known inapproximablity result of the set cover problem.

Theorem 8 ([4]). *The minimum set cover problem with the instance $(U, \mathcal{F})$ is inapproximable within $(1-\epsilon)\ln|U|$ for any $\epsilon > 0$, unless $NP \subset DTIME(n^{\log\log n})$.*

Using the above result, we show that the $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ problem is inapproximable within $\Omega(\lg|\mathcal{A}|)$ unless $NP \subset DTIME(n^{\log\log n})$.

Theorem 9. *The problem of computing the optimum player weights is inapproximable within $(1-\epsilon)\ln\left(\frac{|\mathcal{A}|}{c}\right) + \frac{\ln\left(\left(\frac{|\mathcal{A}|}{c}\right)^{(1-\epsilon)}\right)-1}{N^*}(c-1)$, for any $\epsilon > 0$ (and hence inapproximable within $\Omega(\lg|\mathcal{A}|)$), unless $NP \subset DTIME(n^{\log\log n})$, where N^* denotes the sum of the optimum player weights, and $c = 2$ for eavesdropping adversary and $c = 3$ if the adversary is Byzantine.*

Algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$

1. Given the instance $I_k = (\mathcal{P}, \mathcal{A}, c, k, k)$ of the OAPW problem, construct the instance $I_1 = (\mathcal{P}^{(k)}, \mathcal{A}, c, k, 1)$ of the OAPW problem as illustrated in the proof of Theorem 6.
2. Solve the instance I_1 using the approximate algorithm given in Fig. 1.

Fig. 2. The Approximate Algorithm for $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$

Proof: Given a set $\mathcal{U}$ and a set $\mathcal{F}$ containing subsets of $\mathcal{U}$, the *Set Cover problem* is to find the minimum number of sets of $\mathcal{F}$ that covers $\mathcal{U}$. We show that the Set Cover problem could be solved using an algorithm for the $OAPW$ problem. Given $\mathcal{U}$ and $\mathcal{F}$, construct the instance $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ as follows:

1. Construct a bipartite graph $H = (X, Y, E)$ such that $X = \mathcal{F}$ and $Y = \mathcal{U}$ and $(x, y) \in E$ if and only if $x \in X, y \in Y$ and $y \notin x$.
2. Let $U = \{u_1, u_2, \ldots u_{|\mathcal{U}|}\}$. Construct the set $\mathcal{A}'$ of $|\mathcal{U}|$ elements, such that the i^{th} element of $\mathcal{A}'$, is the set of elements in X that are adjacent to u_i in H. Let $\mathcal{P}' = X$.
3. Let $\mathcal{P}$ be c copies of $\mathcal{P}'$ and $\mathcal{A}$ be c copies of $\mathcal{A}'$.

From the Theorems 2 and 4, it is straightforward to observe that a solution to the instance $OAPW(\mathcal{P}, \mathcal{A}, c, k, k)$ gives a set cover of size $k+1$, and we minimize k to get the Minimum Set Cover.

Let $N^* = ck^*+1$ be the size of the optimal solution to $OAPW(\mathcal{P}, \mathcal{A}, c, k^*, k^*)$ and let $N = ck + 1$ be the size of the solution yielded by our algorithm. This implies that the minimum set cover is of size $k^* + 1$ and the solution got by application of our algorithm is a set cover of size $k + 1$.

From Theorem 8 we see that,

$$k + 1 > (k^* + 1)R,$$

where $R = (1 - \epsilon) \ln \frac{|\mathcal{A}|}{c}$.

Note that $|\mathcal{U}| = \frac{|\mathcal{A}|}{c}$. Therefore we get,

$$\frac{N - 1 + c}{c} > \frac{N^* - 1 + c}{c} R,$$

and thus,

$$\frac{N}{N^*} > R + \frac{(R - 1)(c - 1)}{N^*}$$

Hence the proof. ∎

7 Conclusion

The bottleneck in secure distributed protocols is, in general, the communication/round complexity rather than the computation complexity. In the weighted threshold adversary setting, the communication complexity of the (resulting) protocol can be improved by two (independent) methods, viz., optimizing the players' weights, and developing/adapting the techniques of the threshold setting to the weighted threshold setting. In this work, we studied the former method and examined its complexity. We also presented an approximation algorithm for the Optimal Assignment of Player Weights ($OAPW$) problem and proved an inapproximability bound using the well-known set cover problem. Analyzing the quality of approximation is left open and is attempted in the full version of this paper.

References

1. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of 20th ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1988.
2. R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
3. G. Dobson. Worst-case analysis of greedy heuristics for integer programming with non-negative data. *Math. Oper. Res.*, 7:515–531, 1982.
4. U. Feige. A threshold of $\ln n$ for approximating set cover. In *In Proceedings of 28th ACM Symposium on Theory of Computing (STOC)*, pages 314–318, 1996.
5. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
6. O. Goldreich. Secure multiparty computation, 1998. First draft available at http://theory.lcs.mit.edu/˜ oded.
7. M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multiparty computation. In *16th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 25–34, August 1997.
8. M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, April 2000.
9. Martin Hirt and Ueli Maurer. Robustness for free in unconditional multi-party computation. In *CRYPTO '01*, Lecture Notes in Computer Science (LNCS). Springer-Verlag, 2001.
10. S. Micali and P. Rogaway. *Secure Computation: The information theoretic case*, 1998. Former version: Secure Computation, In *Advances in Cryptology CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 392-404, Springer-Verlag, 1991.