ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Collaboration of Data Using M-Privacy

Thanjai Bharathi[1] A.Karthikeyan[2]

Department of CSE, Arunai Engineering College, Thiruvannamalai, Tamilnadu, India

Department of CSE, Arunai Engineering College, Thiruvannamalai, Tamilnadu, India.

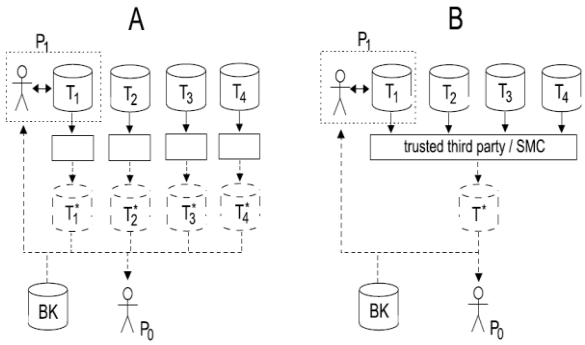
ABSTRACT— We consider the collaborative data publishing problem for anonymizing horizontally partitioned data at multiple data providers. We consider a new type of "insider attack" by colluding data providers who may use their own data records (a subset of the overall data) in addition to the external background knowledge to infer the data records contributed by other data providers. The paper addresses this new threat and makes several contributions. First, we introduce the notion of m-privacy, which guarantees that the anonymized data satisfies a given privacy constraint against any group of up to m colluding data providers. Second, we present heuristic algorithms exploiting the equivalence group monotonicity of privacy constraints and adaptive ordering techniques for efficiently checking m-privacy given a set of records. Finally, we present a data provider-aware anonymization algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy of anonymized data with efficiency. Experiments on real-life datasets suggest that our approach achieves better or comparable utility and efficiency than existing and baseline algorithms while providing m-privacy guarantee.

I. INTRODUCTION

There is an increasing need for sharing data that contain personal information from distributed databases. . For example, in the healthcare domain, a national agenda is to develop the Nationwide Health Information Network (NHIN) to share information among hospitals and other providers, and support appropriate use of health information beyond direct patient care with privacy protection. Privacy preserving data analysis and data publishing, have received considerable attention in recent years as promising approaches for sharing

Fig. 1. Distributed data publishing settings.

data while preserving individual privacy. When the data are distributed among multiple data providers or



data owners, two main settings are used for anonymization. One approach is for each provider to anonymize the data independently (anonymized and-aggregate), which results in potential loss of integrated data utility. A more desirable approach is collaborative data publishing which anonymizes data from all providers as if they would come from one source (aggregate- and-anonymize), using either a trusted third-party (TTP) or Secure Multi-party Computation (SMC) protocols to do computation.

Problem Settings: We consider the collaborative data publishing setting (Figure 1B) with horizontally partitioned data across multiple data providers, each contributing a subset of records *Ti*. As a special case, a data provider could be the data owner itself who is contributing its own records. This is a very common

scenario in social networking and recommendation systems. Our goal is to publish an anonymized view of the integrated data such that a data recipient including the data providers will not be able to compromise the privacy of the individual records provided by other parties. Considering different types of malicious users and information they can use in attacks, we identify three main categories of attack scenarios. While the first two are addressed in existing work, the last one receives little attention and will be the focus of this paper.

A.Attacks by External Data Recipient Using Anonymized Data. A data recipient, e.g. P_0, could be an attacker and attempts to infer additional information about the records using the published data (T^*) and some background knowledge (BK) such as publicly available external data. Most literature on privacy preserving data publishing in a single provider setting considers only such attacks. Many of them adopt a weak or relaxed *adversarial* or *Bayes-optimal privacy* notion to protect against specific types of attacks by assuming limited background knowledge. For example, *k*-anonymity prevents identity disclosure attacks by requiring each equivalence group, records with the same quasi-identifier values, to contain at least *k* records. Representative constraints that prevent attribute disclosure attacks include *l*-diversity, which requires each equivalence group to contain at least *l* "well-represented" sensitive values, and *t*-closeness, which requires the distribution of a sensitive attribute in any equivalence group to be close to its distribution in the whole population

*B.Attacks by Data Providers Using Intermediate Results and Their Own Data***:** We assume the data providers are semi honest, commonly used in distributed computation setting. They can attempt to infer additional information about data coming from other providers by analyzing the data received during the anonymization. A trusted third party (TTP) or Secure Multi-Party Computation (SMC) protocols (e.g.) can be used to guarantee there is no disclosure of *intermediate* information *during* the anonymization.

*C.Attacks by Data Providers Using Anonymized Data and Their Own Data***.** Each data provider, such as P_1 in Figure 1, can also use anonymized data T^* and his own data (T_1) to infer additional information about other records. Compared to the attack by the external recipient in the first attack scenario, each provider has additional data knowledge of their own records, which can help with the attack. This issue can be further worsened when multiple data providers collude with each other. In the social network or recommendation setting, a user (having an account herself) may attempt to infer private information about other users using the anonymized data or recommendations assisted by some background knowledge and her own account information. Malicious users may collude or even create artificial accounts as in a shilling attack. We define and address this new type of "insider attack" by data providers in this paper..

T_1

Name	Age	Zip	Disease
Alice	24	98745	Cancer
Bob	35	12367	Asthma
Emily	22	98712	Asthma

T_2

Name	Age	Zip	Disease
Dorothy	38	98701	Cancer
Mark	37	12389	Flu
John	31	12399	Flu

T_3

Name	Age	Zip	Disease
Sara	20	12300	Epilepsy
Cecilia	39	98708	Flu

T_4

Name	Age	Zip	Disease
Olga	32	12337	Cancer
Frank	33	12388	Asthma

T_a^*

Provider	*Name*	Age	Zip	Disease
P_1	*Alice*	[20-30]	*****	Cancer
P_1	*Emily*	[20-30]	*****	Asthma
P_3	*Sara*	**[20-30]**	*****	**Epilepsy**
P_1	*Bob*	[31-35]	*****	Asthma
P_2	*John*	[31-35]	*****	Flu
P_4	*Olga*	[31-35]	*****	Cancer
P_4	*Frank*	[31-35]	*****	Asthma
P_2	*Dorothy*	[36-40]	*****	Cancer
P_2	*Mark*	[36-40]	*****	Flu
P_3	*Cecilia*	[36-40]	*****	Flu

T_b^*

Provider	*Name*	Age	Zip	Disease
P_1	*Alice*	[20-40]	*****	Cancer
P_2	*Mark*	[20-40]	*****	Flu
P_3	*Sara*	[20-40]	*****	Epilepsy
P_1	*Emily*	[20-40]	987**	Asthma
P_2	*Dorothy*	[20-40]	987**	Cancer
P_3	*Cecilia*	[20-40]	987**	Flu
P_1	*Bob*	[20-40]	123**	Asthma
P_4	*Olga*	[20-40]	123**	Cancer
P_4	*Frank*	[20-40]	123**	Asthma
P_2	*John*	[20-40]	123**	Flu

TABLE I
m-ADVERSARY AND m-PRIVACY EXAMPLE.

II. RELATED WORK AND *m*-PRIVACY DEFINITION

Privacy preserving data analysis and publishing has received considerable attention in recent years [1], [2], [3]. Most work has focused on a single data provider setting and considered the data recipient as an attacker. A large body of literature [2] assumes limited background knowledge of the attacker and defines privacy using relaxed *adversarial* notion [7] by considering specific types of attacks. Representative principles include *k*-anonymity [8], [9], *l*-diversity [7], and *t*-closeness [10]. Few recent works have modeled the instance level background knowledge as corruption and studied perturbation techniques under these weak privacy notions [12]. In the distributed setting we studied, since each data holder knows its own records, the *corruption* of records is an inherent element in our attack model and is further complicated by the collusive power of the data providers. On the other hand, differential privacy [1], [3] is an unconditional privacy guarantee for statistical data release or data computations. While providing a desirable unconditional privacy guarantee, non-interactive data release with differential privacy remains an open problem. Many different anonymization algorithms have been introduced so far including Datafly [13], Incognito [14], and Mondrian [11]. In our research we considered the Mondrian algorithm as a baseline because its efficiency and extensibility. There are some works focused on anonymization of distributed data. [5], [6], [15]

studied distributed anonymization for vertically partitioned data using *k*-anonymity. Zhong etal. [16] studied classification on data collected from individual *data owners* (each record is contributed by one data owner) while maintaining *k*-anonymity. Jurczyk et al. [17] proposed a notion called l^{+}-site-diversity to ensure anonymity for data providers in addition to privacy of the data subjects. Mironov etal. [18] studied SMC techniques to achieve differential privacy. Mohammed et al. [4] proposed SMC techniques for anonymizing distributed data using the notion of *LKC*privacy to address high dimensional data. Our work is the first that considers data providers as potential attackers in the collaborative data publishing setting and explicitly models the inherent instance knowledge of the data providers as well as potential collusion between them for any weak privacy.

We first formally describe our problem setting. Then we present our *m*-privacy definition with respect to a given privacy constraint to prevent inference attacks by *m*-adversary, followed by its properties.

Let $T = \{t_1, t_2, \ldots\}$ be a set of records horizontally distributed among *n* data providers $P = \{P_1, P_2, \ldots P_n\}$, such that $T_i \_ T$ is a set of records provided by P_i. We assume A_S is a sensitive attribute with domain D_S. If the records contain multiple sensitive attributes then a new sensitive attribute A_S can be defined as a Cartesian product of all sensitive attributes. Our goal is to publish an anonymized table T^* while preventing any *m*-adversary from inferring A_S for any single record.

A. m-Privacy

To protect data from external recipients with certain background knowledge *BK*, we assume a given privacy requirement *C*, defined by a conjunction of privacy constraints: $C_1 \wedge C_2 \wedge \ldots \wedge C_w$. If a set of records T^* satisfies *C*, we say $C(T^*) = true$. Any of the existing privacy principles can be used as a component constraint.

In our example (Table I), the privacy constraint *C* is defined as $C = C_1 \wedge C_2$, where C_1 is *k*-anonymity with $k = 3$, and C_2 is *l*-diversity with $l = 2$. Both anonymized tables, T^{*a} and T^{*b} satisfies *C*, although as we have shown earlier, T^{*a} may be compromised by an *m*-adversary such as P_1. We now formally define a notion of *m*-privacy with respect to a privacy constraint *C*, to protect the anonymized data against *m*-adversaries in addition to the external data recipients. The notion explicitly models the inherent data knowledge of an *m*-adversary, the data records they jointly contribute, and requires that each equivalence group, excluding *any* of those records owned by an *m*-adversary, still satisfies *C*.

Definition 2.1: (*m*-PRIVACY) Given *n* data providers, a set of records *T*, and an anonymization mechanism *A*, an *m*-adversary *I* ($m \leq n$ - 1) is a coalition of *m* providers, which jointly contributes a set of records T_I. Sanitized records $T^* = A(T)$ satisfy *m*-privacy, i.e. are *m*-private, with respect to a privacy constraint *C*.

B. Monotonicity of Privacy Constraints

Generalization based monotonicity has been defined for privacy constraints in the literature (Definition 2.2) [7], [10] and has been used for designing efficient generalization algorithms to satisfy a privacy constraint ([9], [11], [7], [10]). In this paper we will refer to it as *generalization monotonicity*.

Definition 2.2: (GENERALIZATION MONOTONICITY OF A PRIVACY CONSTRAINT [7], [10]) A privacy constraint *C* is generalization monotonic if and only if for any set of anonymized records T^* satisfying *C*, all its further generalizations satisfy *C* as well.

Generalization monotonicity assumes that original records *T* have been already anonymized and uses them for further generalizations. In this paper, we also introduce more general, record-based definition of monotonicity in order to facilitate the analysis and design of efficient algorithms for checking *m*-privacy.

III. VERIFICATION OF *m*-PRIVACY

Checking whether a set of records satisfies *m*-privacy creates a potential computational challenge due to the combinatorial number of *m*-adversaries that need to be checked. In this section, we first analyze the problem by modeling the checking space. Then we present heuristic algorithms with effective pruning strategies and adaptive ordering techniques for efficiently checking *m*-privacy for a set of records w.r.t. an EG monotonic privacy constraint *C*.

A. Adversary Space Enumeration

Given a set of $_{nG}$ data providers, the entire space of *m*adversaries (*m* varying from 0 to n_G - 1) can be represented using a lattice shown in Figure 2. Each node at layer *m* represents an *m*-adversary of a particular combination of *m*-providers. The number of all possible *m*-adversaries is equal to $(^{n}{}_{m^g})$. Each node has parents (children) representing their direct super-(sub-) coalitions. For simplicity the space is also represented as a *diamond*, where a horizontal line corresponds to all *m*-adversaries with the same *m* value, the bottom node corresponds to 0-adversary (external data recipient), and the top line to (n_G - 1)-adversaries. In order to verify *m*-privacy w.r.t. a constraint *C* for a set of records, we need to check *C* for the records excluding any subset of records owned by any *m*-adversary.

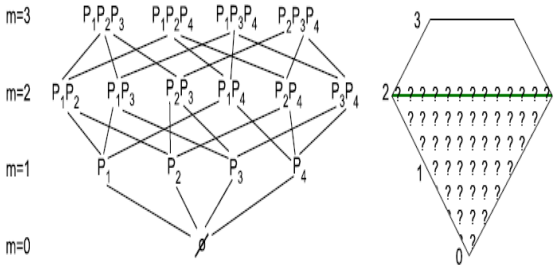


Fig. 2. *m*-Adversary space.

When *C* is EG monotonic, we only need to check *C* for the records excluding all records from any *m*-adversary. For example, in Figure 2, given $m = 2$, all coalitions that need to be checked are represented by question marks.

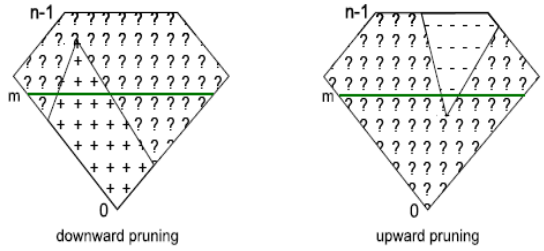
B. Heuristic Algorithms

The key idea of our heuristic algorithms is to efficiently search through the adversary space with effective pruning such that not all *m*-adversaries need to be checked. This is achieved by two different pruning strategies, an adversary ordering technique, and a set of search strategies that enable fast pruning.

C.Pruning Strategies: The pruning strategies are possible thanks to the EG monotonicity of *m*-privacy. If a coalition is not able to breach privacy, then all its sub-coalitions will not be able to do so and hence do not need to be checked (downward pruning).

On the other hand, if a coalition is able to breach privacy, then all its super-coalitions will be able to do so and hence do not need to be checked (upward pruning). In fact, if a sub-coalition of an *m*-adversary is able to breach privacy, then the upward pruning allows the algorithm to terminate

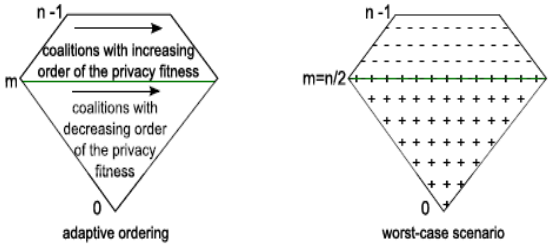
Fig. 3. Pruning strategies for *m*-privacy check



immediately as the *m*-adversary will be able to breach privacy (*early stop*). Figure 3 illustrates the two pruning strategies where + represents a case when a coalition does not breach privacy and - otherwise.

D.Adaptive Ordering of Adversaries: In order to facilitate the above pruning in both directions, we adaptively order the coalitions based on their attack

powers (Figure 4). This is motivated by the following observations. For downward pruning, super-coalitions of *m*-adversaries with limited attack powers are preferred to check first as they are less likely to breach privacy and hence increase the chance of downward



pruning.

In contrast, sub-coalitions of *m*-adversaries with significant attack powers are preferred to check as they are more likely to breach privacy and hence increase the chance of upward pruning (early-stop).

Fig. 4. Adaptive ordering for efficient pruning and the worst-case scenario without any pruning possible

To quantify privacy fulfillment by a set of records, which is used to measure the attack power of a coalition and privacy of remaining records (used to facilitate the anonymization, which we will discuss in next section), we introduce the privacy fitness score w.r.t. *C* for a set of records.

Definition 3.1: (PRIVACY FITNESS SCORE) Privacy fitness F_C for a set of records T^* is a level of the fulfillment of the privacy constraint *C*. A privacy fitness score is a function *f* of privacy fitness with values greater or equal to 1 only if $C(T^*) = true$,

$$score_{F_C(T^*)} = f(F_{C1}(T^*), F_{C2}(T^*), \ldots, F_{Cw}(T^*))$$

In our setting, *C* is defined as *k*-anonymity ^*l*-diversity. The privacy fitness score can be defined as a weighted average of the two fitness scores with $\alpha \in (0, 1)$. When $C(T^*) = false$, $score_{F_C(T^*)} = max(1-\epsilon, F_C(T^*))$, where ϵ is small.

In our example $score_{F_C}$ is defined as follow:

$$score_{F_{C_1 \wedge C_2}}(T^*) = (1-\alpha)\cdot\frac{|T^*|}{k} + \alpha\cdot\frac{|\{t[A_S] : t \in T^*\}|}{l} \quad (1)$$

In order to maximize the benefit of both pruning strategies, the super-coalitions of *m*-adversaries are generated in the order of ascending fitness scores (ascending attack powers), and the sub-coalitions of *m*-adversaries are generated in the order of descending fitness scores (descending attack powers) (Figure 4).

Algorithm 1: The *binary* verification algorithm

```
Data: A set of records T provided by P_1, ..., P_{n_G}, a monotonic privacy
      constraint C, a privacy fitness scoring function score_F and the m value
Result: true if T* is m-private, false otherwise
1  begin
2      sites = sort_sites(P, increasing_order, score_F)
3      use_adaptive_order_generator(sites, m)
4      while is_m-privacy_verified(T*, m) = false do
5          I_super = next_coalition_of_size(n − 1)
6          if privacy_is_breached_by(I_super) then
7              continue
8          I_sub = next_sub-coalition_of(I_super, m)
9          if privacy_is_breached_by(I_sub) then
10             return false //early stop
11         while is_coalition_between(I_sub, I_super) do
12             I = next_coalition_between(I_sub, I_super)
13             if privacy_is_breached_by(I) then
14                 I_super = I
15             else
16                 I_sub = I
17         prune_all_sub-coalitions(I_sub)
18         prune_all_super-coalitions(I_super)
19     return true
```

Now we present several heuristic algorithms that use different search strategies, and hence utilize different pruning. All of them use the adaptive ordering of adversaries to enable fast pruning.

E.The Top-Down Algorithm: The *top-down* algorithm checks the coalitions in a top-down fashion using downward pruning, starting from (n_G -1)-adversaries and moving down until a violation by an *m*-adversary is detected or all *m*-adversaries are pruned or checked.

F.The Bottom-Up Algorithm: The *bottom-up* algorithm checks coalitions in a bottom up fashion using upward pruning, starting from 0-adversary and moving up until a violation by any adversary is detected (*early-stop*) or all *m*-adversaries are checked.

*G.The Binary Algorithm***:** The *binary* algorithm, inspired by the binary search algorithm, checks coalitions between (n_G - 1) -adversaries and *m*-adversaries and takes advantage of both upward and downward pruning's (Figure 5, Algorithm 1). The goal of each iteration is to search for a pair I_{sub} and I_{super}, such that I_{sub} is a direct sub-coalition of I_{super} and I_{super} breaches privacy while I_{sub} does not. Then I_{sub} and all its sub-coalitions are pruned (downward pruning), I_{super} and all its super-coalitions are pruned (upward pruning) as well.

IV. Anonymization for *m*-Privacy

After defining the *m*-privacy verification algorithm, we can now use it in anonymization of a horizontally distributed dataset to achieve *m*-privacy. In this section, we will present a baseline algorithm, and then our approach that utilizes a data provider-aware algorithm with adaptive *m*-privacy checking strategies to ensure high utility and *m*-privacy for anonymized data.

Since we have shown that *m*-privacy with respect to a generalization monotonic constraint is generalization monotonic (Theorem 2.1), most existing generalization-based anonymization algorithms can be modified to achieve *m*-privacy – every time a set of records is tested for a privacy constraint *C*, we check *m*-privacy w.r.t. *C* instead. As a baseline algorithm to achieve *m*-privacy, we adapted the multidimensional Mondrian algorithm [11] designed for *k*-anonymity. A main limitation of such a simple adaptation is that groups of records are formed *oblivious* of the data providers, which may result in over-generalization in order to satisfy *m*-privacy.

Algorithm 2: The *provider-aware* algorithm.

```
Data: A set of records T = ⋃_{j=1}^{n} T_j provided by {P_1, P_2, ..., P_n}, a set of
      QI attributes A_i (i = 1, ..., q), m, a privacy constraint C
Result: Anonymized T* that satisfies m-privacy w.r.t. C
1  begin
2      π = get_splitting_points_for_attributes(A_i)
3      π = π∪ get_splitting_point_for_providers(A_0)
4      π' = {a_i ∈ π, i ∈ {0, 1, ..., q} :
       are_both_split_subpartitions_m-private(T, a_i)}
5      if π' is ∅ then
6          T* = T*∪ generalize_all_QIs(T)
7          return T*
8      A_j = choose_splitting_attribute(T, C, π')
9      (T'_r, T'_l) = split(T, A_j)
10     Run recursively for T'_l and T'_r
```

We introduce a simple and general algorithm based on the Binary Space Partitioning (BSP) Algorithm 2). Similar to the Mondrian algorithm, which is also an example of BSP algorithms, it recursively chooses an attribute to split data points in the multidimensional domain space until the data cannot be split any further while satisfying *m*-privacy w.r.t. *C*. However, the algorithm has three novel features: 1) it takes into account the data provider as an additional dimension for splitting; 2) it uses the privacy fitness score as a general scoring metric for selecting the split point; 3) it adapts its *m*-privacy verification strategy for efficient verification. The pseudo code for our *provider-aware* anonymization algorithm is presented in Algorithm 2. We describe the algorithm details with respect to the novel features below.

V. Experimental Results

We present two sets of experiment results with the following goals: 1) to compare and evaluate the different *m*-privacy verification algorithms given a set of records, and 2) to evaluate and compare the proposed anonymization algorithm for a given dataset with the baseline algorithm in terms of both utility and efficiency.

A. Experiment Setup

We used combined training and test sets of the Adult dataset2. Records with missing attribute values have been removed. All remaining 45,222 records have been used in all experiments. The *Occupation* has been chosen as a sensitive attribute *As*. This attribute has 14 distinct values. Data are distributed among n data providers $P_1, P_2, \ldots, P_n$ such that their distribution follows a uniform or exponential distribution. We observe similar results for both of them and only report those for the exponential distribution in the paper.

2The Adult dataset has been prepared using the Census database from 1994, http://archive.ics.uci.edu/ml/datasets/Adult

The privacy constraint C is defined by k-anonymity [9] and l-diversity [7]. C is EG monotonic. We note again m-privacy is orthogonal to the privacy constraint being used in its definition. Both m-privacy verification and anonymization use privacy fitness scores, but with different values of the weight parameter α. Values of α can be defined in a way that reflects restrictiveness of privacy constraints.

The impact of the weight parameter to overall performance was experimentally investigated and values of α for the most efficient runs have been chosen as defaults. All experiment and algorithm parameters and their default values are listed in Table II.

Name	Description	Verification	Anonymization
α	Weight paramter	0.3	0.8
m	Power of m-privacy	5	3
n	Total number of data providers	–	10
n_G	Number of data providers contributing to a group	15	–
$\|T\|$	Total number of records	–	45,222
$\|T_G\|$	Number of records in a group	{150, 750}	–
k	Parameter of k-anonymity	50	30
l	Parameter of l-diversity	4	4

TABLE II
EXPERIMENT PARAMETERS AND DEFAULT VALUES.

B. m-Privacy Verification:

The objective of the first set of experiments is to evaluate the efficiency of different algorithms for m-privacy verification given a set of records T_G with respect to the previously defined privacy constraint C.

Attack Power: In this experiment, we compared the different m-privacy verification heuristics against different attack powers. We used two different groups of records with relatively small and large average number of records per data provider, respectively. Figure 6 shows the runtime with varying m for different heuristics for the two groups.

The first group counts 150 records and has a small average fitness score per provider (equal to 0.867), which reflects a high probability of privacy breach by a large m-adversary. For almost all values of m the *binary* algorithm achieves the best performance due to its efficient upward and downward pruning. However, the *top-down* algorithm is comparable with *binary* for $m > n_G/2$.

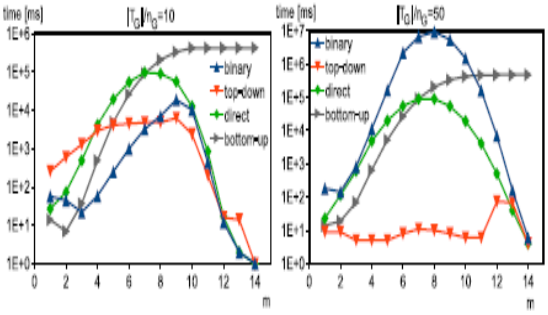


Fig. 6. Runtime (logarithmic scale) vs. m.

The second group counts 750 records and has a larger average fitness score per provider (equal to 2.307). Therefore intuitively, it is very unlikely that a coalition of adversaries will be able to breach privacy and the downward pruning can be applied often. This intuition is confirmed by results, which show that the *top-down* algorithm is significantly better than other heuristics. Since the remaining algorithms do not rely so much on the downward pruning, they have to perform an exponential number of checks. We can also observe a clear impact of m when $m \approx n_G/2$ incurs the highest cost.

VI. CONCLUSIONS

In this paper, we considered a new type of potential attackers in collaborative data publishing – a coalition of data providers, called m-adversary. To prevent privacy disclosure by any m-adversary we showed that guaranteeing m-privacy is enough. We presented heuristic algorithms exploiting equivalence group monotonicity of privacy constraints and adaptive ordering techniques for efficiently checking m-privacy. We introduced also a *provider-aware* anonymization algorithm with adaptive m-privacy checking strategies to ensure high utility and m-privacy of anonymized data. Our experiments confirmed that our approach achieves better or comparable utility than existing algorithms while ensuring m-privacy efficiently. There are many remaining research questions. Defining a proper privacy fitness score for different privacy constraints is one of them. It also remains a question to address and model the data knowledge of data providers when data are distributed in a vertical or ad-hoc fashion. It would be also interesting to verify if our methods can be adapted to other kinds of data such as set-valued data.

REFERENCES

[1] C. Dwork, "Differential privacy: a survey of results," in *Proc. of the 5th Intl. Conf. on Theory and Applications of Models of Computation*, 2008, pp. 1–19.

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, pp. 14:1–14:53, June 2010.

[3] C. Dwork, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, pp. 86–95, January 2011.
[4] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. Lee, "Centralized and distributed anonymization for high-dimensional healthcare data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 4, pp. 18:1–18:33, October 2010.
[5] W. Jiang and C. Clifton, "Privacy-preserving distributed k-anonymity," in *Data and Applications Security XIX*, ser. Lecture Notes in Computer Science, 2005, vol. 3654, pp. 924–924.
[6] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," *VLDB J.*, vol. 15, no. 4, pp. 316–333, 2006.
[7] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *ICDE*, 2006, p. 24.
[8] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE T. Knowl. Data En.*, vol. 13, no. 6, pp. 1010–1027, 2001.
[9] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J.Uncertain. Fuzz.*, vol. 10, no. 5, pp. 557–570, 2002.
[10] N. Li and T. Li, "t-closeness: Privacy beyond k-anonymity and ldiversity," in *In Proc. of IEEE 23rd Intl. Conf. on Data Engineering (ICDE)*, 2007.
[11] K. Lefevre, D. J. Dewitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *ICDE*, 2006.
[12] Y. Tao, X. Xiao, J. Li, and D. Zhang, "On anti-corruption privacy preserving publication," in *Proc. of the 2008 IEEE 24th Intl. Conf. on Data Engineering*, 2008, pp. 725–734.
[13] L. Sweeney, "Datafly: A system for providing anonymity in medical data," in *Proc. of the IFIP TC11 WG11.3 Eleventh Intl. Conf. on Database Security XI: Status and Prospects*, 1998, pp. 356–381.
[14] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in *Proc. of the 2005 ACM SIGMOD Intl.Conf. on Management of Data*, 2005, pp. 49–60.
[15] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung, "Privacypreserving data mashup," in *Proc. of the 12th Intl. Conf. on Extending Database Technology*, 2009, pp. 228–239.
[16] S. Zhong, Z. Yang, and R. N. Wright, "Privacy-enhancing kanonymization of customer data," in *Proc. of the 24th ACM SIGMODSIGACT-SIGART Symposium on Principles of Database Systems*, 2005, pp. 139–147.
[17] P. Jurczyk and L. Xiong, "Distributed anonymization: Achieving privacy for both data subjects and data providers," in *DBSec*, 2009, pp. 191–207.
[18] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan, "Computational differential privacy," in *Advances in Cryptology – CRYPTO 2009*, ser. Lecture Notes in Computer Science, vol. 5677, 2009, pp. 126–142.