This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Distributed Data Mining with Differential Privacy

Ning Zhang, Ming Li, Wenjing Lou
Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, MA
Email: {ning, mingli}@wpi.edu, wjlou@ece.wpi.edu

***Abstract*—With recent advances in communication and data storage technology, an explosive amount of information is being collected and stored in the Internet. Even though such vast amount of information presents great opportunities for knowledge discovery, organizations might not want to share their data due to legal or competitive reasons. This posts the challenge of mining knowledge while preserving privacy. Current efficient privacy-preserving data mining algorithms are based on an assumption that it is acceptable to release all the intermediate results during the data mining operations. However, it has been shown that such intermediate results can still leak private information. In this work, we use differential privacy to quantitatively limit such information leak. Differential privacy is a newly emerged privacy definition that is capable of providing strong measurable privacy guarantees. We propose Secure group Differential private Query(SDQ), a new algorithm that combines techniques from differential privacy and secure multiparty computation. Using decision tree induction as a case study, we show that SDQ can achieve stronger privacy than current efficient secure multiparty computation approaches, and better accuracy than current differential privacy approaches while maintaining efficiency.**

I. Introduction

With recent advances in communication and data storage technology, an explosive amount of information is being collected and stored in the Internet. Such vast amount of data provides great opportunities for knowledge discovery. There are many application areas of data mining, such as business logic modeling, disease control, intrusion detection, medical research and etc. With the emergence of such unprecedented amount of information stored at different physical location, distributed data mining has become one of the key enablers of large scale knowledge extraction. However, information is almost always collected under certain privacy considerations. Organizations might not want to share with each other the contents of their data, sometimes even the statistic either due to legal or competition constraints. An example of such conflict is the disease control mining operation. Insurance companies hold great volume of medical claim data which can be used for data mining to determine whether a disease outbreak is occurring or not. However, they might deny participation because of possible legal complications for disclosing their data. The task of performing data mining across multiple data sources to extract the knowledge and not revealing any extra information is usually called privacy-preserving data mining.

Verykios et al. [1] attempted to classify privacy-preserving data mining with five dimensions, data distribution, data modification, data mining algorithm, data or rule hiding and privacy preservation. For our work in this paper, we will be focusing on privacy preservation while learning from sources distributed across the network. Privacy preserving often refers to different techniques of modifying data such that its data mining utility stays the same while satisfying the privacy requirement. Privacy-preserving data mining algorithms with pure secure multiparty computation approach [2] are usually not scalable, therefore a lot of existing efficient algorithms use a definition that assumes it is acceptable to release the intermediate results during the mining operations, while exposing these results can actually compromise individual record privacy [1], [3]–[5].

In this work, we recognize such information leak can be limited quantitatively by incorporating differential privacy into secure multiparty computation. Differential privacy [6]–[8] is a newly emerged definition of privacy for statistical databases whose primary function is answering statistical queries, such as count, sum and etc. Differential privacy essentially mandates the outcome of any operation to be insensitive to change of any particular record in the data set, thus limiting the information leak of an individual record due to queries. By adding a calibrated noise to the intermediate results during the data mining operations, we quantitatively bound the relative change in probability of an event due to a change in a single record [7]. However, a direct adoption of differential privacy on existing efficient privacy-preserving data mining algorithms is weak against collusion attacks. A random select algorithm is thus proposed to randomize the noise addition process in order to reduce the probability of successful collusion attacks. Using decision tree induction as a case study, we demonstrate our approach can obtain comparable accuracy and stronger privacy guarantee than simple use of multiparty computation. In addition, the accuracy of our algorithm is significantly higher than direct adoption of differential privacy.

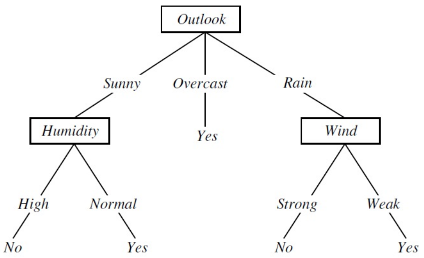
II. Related Work

Privacy-preserving data mining has been an active research area for a decade. Some used randomization techniques to modify data to preserve individual record's privacy [9], [10]. Others used techniques from secure multiparty computation to protect privacy of individual record and the database [3], [4], [11], [12].

Differential privacy is a recent definition of privacy tailored for statistical databases [6], [8], [13]. SuLQ was proposed by Dwork et al. [7], [14] as a primitive to achieve differential privacy by adding calibrated noise to the actual result. McSherry et al. [13] introduced the differential privacy mechanism design.

This work is closely related to [4] and [15]. Emekci et al. [4] used Shamir's secret sharing technique to securely compute the sum of counts across databases. In [15], Friedman compared

978-1-61284-233-2/11/$26.00 ©2011 IEEE



Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Fig. 1. Sample Decision Tree

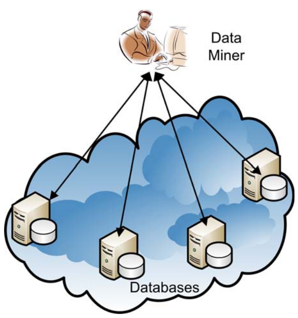


Fig. 2. System Diagram

the effectiveness of various differential privacy primitives in data mining, specifically building classifiers.

III. Background

A. Decision Tree - Distributed ID3 Algorithm

Decision tree is one of the most widely used classification methods in data mining. We will use it as a case study for this work. Shown in Fig. 1 [12] is an example of decision tree induction problem. The goal is to predict whether to go out to play or not for a future day with certain feature values using patterns learned from existing data.

In a decision tree, one start with the root node, and split on the attribute with highest quality value, until a classification can be made. ID3 is a decision tree induction algorithm that uses information gain as the quality function for choosing attributes. Information gain is defined as the difference of entropy of data set T before and after it is splitted with attribute A, $InfoGain(A) = E(T) - E(T \mid A)$. If there are l categories $ci...c_l$, T_{c_i} is the set of records where class $= c_i$, and $\mid T \mid$ is cardinality of the set, then the entropy $E(T)$ is defined as

$$E(T) = \sum_{i=1}^{l} (-\frac{\mid T_{c_i} \mid}{\mid T \mid} \cdot log\frac{\mid T_{c_i} \mid}{\mid T \mid})$$

To make a classification, one can simply mark the leaf node with the class that has the highest number of instances. This method of inducing decision tree can be easily extended to the distributed environment. The count we used in single source is simply the sum of counts from set of source now.

B. Secure Multiparty Computation

Two or more parties would like to perform some computation with their secret value but none of them wants to disclose its value to others is a typical problem for Secure Multiparty Computation (SMC). We will be referring to the basic techniques of secret sharing [3], [16], secure equality test [17], [18] and secure sum [1]–[3], [17] through out the paper.

C. Differential Privacy

Differential privacy is a new way to define privacy quantitatively for statistical databases [6], [8]. A randomized function M gives ϵ-differential privacy if for all data sets A and B differing on at most one element, and $S \subseteq Range(M)$,

$$\Pr[M(A) \in S] \leq exp(\epsilon) \times \Pr[M(B) \in S]$$

ϵ can be think of as the degree of the privacy guarantee, the smaller the value of ϵ, the stronger the privacy guarantee it provides. There are several differential privacy primitives [6], [7], [13]. For count query f, the mechanism $M(X) = f(X) + Laplace(1/\epsilon)$ maintains ϵ-differential privacy [7]. Note that the amount of noise added depends only on sensitivity of the function and the privacy control parameter ϵ, therefore if $f(X)$ is really large, the amount of noise is relatively small.

When multiple operations are performed on the same set of data, it is called sequential composition [19]. It is natural that privacy degrades with sequential composition since more information is exposed. However differential privacy has the advantage that privacy degrades in a well controlled manner. A sequence of query with ϵ_i-differential privacy provides $\sum_i \epsilon_i$-differential privacy [15], [19]. Since decision tree induction is a sequence of queries by the data miner, the sum can be considered as the privacy budget for the mining operation.

IV. Problem Formulation

A. System Model

The system is a distributed environment consisting of N databases across the network. Each database is individually the owner of its data, therefore all the data is stored in plaintext form. Data is horizontally distributed among N statistical databases. As shown in Fig. 2, the parties involved are the individual databases and data miner. The data miner induces decision tree by issuing various count queries. We assume the communication links between parties are secured by existing techniques.

B. Design Goals

There are two types of counts generated from the queries during distributed tree induction: the count from individual database and the sum of counts, i.e., the aggregate count from all databases. Exposing any of them will lead to information leak. We aim to protect the privacy of both types of counts.

1) Privacy of Individual Count: Exact count of records in every individual database as a result for each query operation should be confidential against other parties. If the count is not protected, sensitive information of both the site holding the database and individual record owner will be leaked. An example of the former is, if insurance company A find outs the exact count of all patients in Boston for insurance company B, A can utilize this information to tailor its strategy in Boston. However B may consider this statistic information private and may not want to share it with any other organizations. For the latter, an example would be a count query with result equal to exactly one exposes all attributes of an individual record [5].

2) Privacy of Aggregate Count: Exact aggregate count should be hidden against all parties. If this value is exposed, privacy of individual records will be compromised using the same inference technique described above. However, data miner relies on the aggregate count to extract knowledge, thus we use differential privacy to resolve this conflict by bounding the probability information leak of the individual records. Therefore, the aggregate count should be a noisy count that maintains differential privacy.

C. Threat Model

The adaptive semi-honest adversary model proposed in [12] will be used in our system. In this adversary model, all parties, even corrupted ones, follow the protocol honestly, but adversary obtains all their internal values. An adversary is adaptive if it is capable of choosing which parties to corrupt during the computation, rather than having a fixed set of corrupted parties. Though this is a relatively weak adversarial model, it has been widely accepted as an appropriate model for distributed data mining [4], [12]. An adversary can be either the miner or a database, who will try to learn both the accurate counts of individual databases and sum of counts from all databases.

V. SOLUTION

In this section, we will first introduce two existing solutions for building decision trees, ID3 algorithm with secure sum and ID3 algorithm with SuLQ, and then our proposed ID3 algorithm with **S**ecure Group **D**ifferential Private **Q**uery, SDQ.

A. Distributed ID3 Direct Secure Sum

The simplest approach to building the distributed ID3 is to have the miner query individual databases directly. However this exposes individual counts and aggregate count, which is not acceptable. Emekci et al. [4] used secure sum primitive to aggregate the individual counts and present only the aggregate count to the miner. Though this approach guarantees confidentiality of individual count, it still exposes the accurate sum to the miner, which compromises the individual record privacy.

B. Distributed ID3 SuLQ

A direct adoption of SuLQ for distributed environment can be modified from the existing single database algorithm in [14], [15]. Each database will act as the differential privacy curator for its own data.

Calibrated noise is added to all queries to maintain the differential privacy. Since each query result from individual database is protected, there is simply no way for others to learn the accurate count. Therefore in this scheme, accurate count is always protected. However, if the goal is to protect only the sum of all counts with ϵ-differential privacy, then sum of all counts calculated by the miner will have a noise with standard deviation that's $\sqrt{N}$ bigger than what's needed to keep the sum ϵ-differential private.

Algorithm 1 Random Select

```
procedure RANDOMSELECT($M, \mathcal{D}$)
Require: Data Miner $M$, Databases $\mathcal{D} = \{D_0, ..., D_{N-1}\}$
    $M$ randomly assign an unique index $I_i \in [0, N)$ for each $D_i$ and splits each $I_i$ using secret sharing scheme into $N$ secret sharing where each $D_i$ holds a secret share of $I_i$
    $\mathcal{D}$ jointly create a random number $R$ using JRP, and each $D_i$ holds a secret share $R_i$
    ModuloReduction($R, N$)
    for every $i = 0 \rightarrow N - 1$ do
        $D_i$ initiates secureEqualityTest($R$ and $I_i$)
        if $D_i$ gets 1 from equality test then $D_i$ is selected
        end if
    end for
end procedure
```

*C. **S**ecure Group **D**ifferential Private **Q**uery*

Recognizing the privacy problem in direct secure sum ID3 and the excessive noise in direct SuLQ ID3, we proposed ID3 with **S**ecure group **D**ifferential private **Q**uery, SDQ, where primitives from differential privacy and secure multiparty computation are combined to give a more accurate aggregated query result of a group of databases while maintaining the same level of differential privacy on the final output.

Shown above, in either method the data miner only needs the final sum of all the counts from each individual databases, either through secure sum or adding up results from direct noisy queries. The ability of direct querying individual database in distributed SuLQ really is not necessary. In that case, we can have a trusted curator add the calibrated noise before secure sum operation to guaranteed the output of secure sum provides ϵ-differential privacy and none of databases learns other's input. The end result will have only one noise drawn from $Lap(1/\epsilon)$ instead of the sum of N noises from the same distribution. While such an approach can provide the same level of differential privacy to the end result, it is weak against collusion. Suppose the data miner knows database D_d is the trust curator that adds noise to the final count. She can corrupt the trusted curator to find out noise added to recover the accurate final count. To reduce the probability of such collusion attack, we propose a random select algorithm in Alg. V-B to have all the databases jointly select the curator securely randomly.

In random select, the data miner assigns each database an unique index randomly, and distributes each the index in N secret share form in Z_p to all individual databases. All databases then jointly creates a random number in Z_p in secret share form, using JRP^q in [17]. The secret share of the random number is converted to modulo N using the modulo reduction technique [17]. Then for every database D_i, it uses secure equality test techniques [17] to compute $R = I_i$. Since we know R has a range of 0 to $N - 1$, and the index for each database is different, therefore R will only match one of I_i. Thus this algorithm randomly picks one database such that only the selected database finds out she is selected while other database gains no knowledge at all. Since none of the databases can gain knowledge of the jointly created random R, the data miner will not be able determine which database is selected unless he happens to corrupt the randomly selected database.

In order to achieve differential privacy on the output of secure sum, a noise drawn from $Lap(1/\epsilon)$ needs to be added into the count before the sum is performed. Since $Lap(1/\epsilon)$ is the difference of two independent exponential random variables [6], we can have two databases D_{+Expo}, D_{-Expo} picked by random select in Alg. V-B to add the positive exponential noise Exponential($1/\epsilon$) and negative exponential noise Exponential($1/\epsilon$) respectively, as shown in Alg. 2. In SDQ ID3, the idea is the same as ID3 tree induction except when counts are needed to calculate entropy gain, it utilizes SDQ to get counts from the group of database.

Algorithm 2 Secure Group Differential Private Query

```
procedure SDQ($M, \mathcal{D}, Q, \epsilon$)
Require: Data Miner $M$, Databases $\mathcal{D}$, Count Query $Q$, Privacy Level $\epsilon$
    $D_{+Expo}$ = randomSelect($M, \mathcal{D}$)
    $D_{-Expo}$ = randomSelect($M, \mathcal{D}$)
    $Q_{P_{+Expo}}$ += Exponential($1/\epsilon$);
    $Q_{P_{-Expo}}$ -= Exponential($1/\epsilon$);
    return SecureSum($\mathcal{D}, Q$) to $M$
end procedure
```

Algorithm 3 SDQ ID3

```
procedure SDQ ID3($M, \mathcal{D}, B, \mathcal{P}, \mathcal{A}, d$)
Require: Data Miner $M$, Databases $\mathcal{D}$, Privacy Budget $B$, Attributes $\mathcal{A}$, tree depth $d$
    $\epsilon$ = calculateBudget($B, d$)
    if reachLeafNode then
        label node $C$=argmax$_c$(SDQ($M, \mathcal{D}, Count, \epsilon$))
    else
        $\forall A \in \mathcal{A}$ calculate InfoGain($A$) with SDQ
        split tree with A that maximize Gain($A$)
    end if
end procedure
```

Remark Even though secure equality and secure modulo reduction are not as efficient as secure sum, they are still much better than existing protocols that rely heavily on cryptographic primitives to securely induce the decision tree [2].

VI. Security Analysis

In this section, we will analyze SDQ ID3 see if it meets the security goals of the system.

A. Without Collusion

1) Privacy of Individual Count: This property is guaranteed by secure sum in SDQ as proved in [3], [4].

2) Privacy of Aggregate Count: This is shown while introducing Alg. 2, SDQ. We add a calibrated noise to the actual count before the secure sum, and since secure sum adds up all secret input accurately, the calibrated noise is added to the final sum accurately which provides ϵ-differential privacy. This ϵ-differential privacy was not available in the ID3 with secure sum. When the adversary is data miner, he only has the noisy count. When the adversary is a database he learns neither the noisy sum nor the actual sum.

B. With Collusion

1) Privacy of Individual Count: This property is guaranteed by secure sum in SDQ as proved in [3], [4] even under collusion up to $N-1$ parties.

2) Privacy of Aggregate Count: When the adversary is the data miner and she is able to collude with the two databases that jointly add the calibrated noise to provide differential privacy, she gains access to the true count. We use random select to bound the probability of a success collusion attack. Since neither the databases nor the miner can know which two databases are selected, the miner can only randomly pick a database to corrupt. Therefore assuming there are N databases, and miner can only corrupt C of them, then probability miner gaining access to true count is $\frac{C\cdot(C-1)}{N\cdot(N-1)}$. If the miner can control 100 out of 1000 databases, he only has around 1% chance learning the true count. When the adversary is a database, he will need to first corrupt the data miner, since only the data miner has the noisy sum from SDQ. Once the miner is corrupted the same analysis for data miner being an adversary applies, however if the database cannot corrupt the miner, he will have to corrupt all other database in order to gain access to accurate aggregate count.

Therefore under our threat model, SDQ is most effective when the number of databases is large, because the relative probability of adversary gaining accurate count of the sum in such a setting is very small.

VII. Experiment

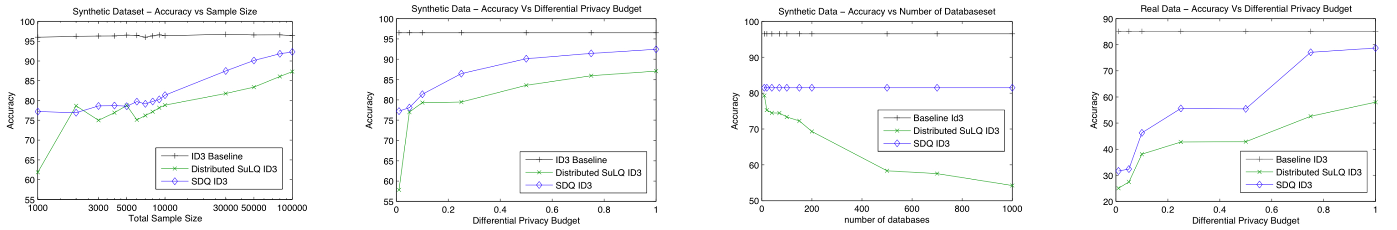
In this section, we will present four experiments designed to evaluate various aspects of SDQ ID3 with synthetic and real data set. We created several instance types on top of Weka [20] data learning package. 100 runs of 10-fold cross-validation is used to evaluate the accuracy of various algorithm.

A. Experiment with Synthetic Data

We generated all of the synthetic data sets using RDG data generator in Weka repository. Noise was added to the data by randomly reassigning values for each attribute of every records 10% of the time. Three experiments were performed with the synthetic data sets to analyze the accuracy of SDQ ID3 with different sample size, security level and the number of databases.

1) Accuracy with Respect to the Sample Size: In this experiment, we set privacy budget to 0.1, number of database to 10 and vary the number of record inside individual database. As we can see from Fig. 3(a), all the algorithms have better accuracy as the sample size grows. However ID3 with SDQ still maintains its advantage at all data sizes.

2) Accuracy with Respect Differential Privacy Budget: The level of differential privacy often depends on the sensitivity of data. Sensitive data will have smaller privacy budget. Our second experiment looks at how various algorithms respond to changes in the differential privacy level. In this experiment, we assume there are 10 databases, each with 1000 records. Shown in Fig. 3(b), the accuracy improves with larger privacy budget. It is because the bigger privacy budget means the lower privacy level for each query, and thus the smaller the noise. Note that SDQ ID3 always outperforms distributed SuLQ ID3. The accuracy



(a) Acc Vs SampleSize, N=10, B=0.1

(b) Acc Vs Privacy, N = 10, total data size = 10000

(c) Acc Vs Num Database, B = 0.1, total data size = 10000

(d) Acc Vs Privacy, N = 10 databases, B = 0.1

Fig. 3. SDQ Experiments

similarity at privacy budget 0.005 can be explained by low tree depth due to excessive noise.

3) Accuracy with Respect to Number of Databases: Since this is a distributed environment, another question would be how well the algorithms perform when data is sparsely located in large number of databases. We conduct the experiment by setting privacy budget to 0.1, total number of samples to 10000, and vary the number of databases. Under this setting, the number of records each database has will be less with more databases. Results shown in Fig. 3(c) matches our expectation. SDQ always aggregates information with exactly one noise, therefore its accuracy remains independent of the number of databases. As data becomes sparser, the noise of summing up all counts from independently differential private databases increase, therefore affecting the overall accuracy of distributed SuLQ.

B. Experiment with Real Data

Though results from synthetic data set experiments look promising, we think an experiment on real world data is necessary to gain insight into how well SDQ applies to real world problems. We choose Nursery data set in UCI data mining repository [21], because its categorical formation is similar to many other applications. With the data set distributed evenly in 10 databases, we induce the decision tree up to 4 levels with various differential privacy budget. Shown in Fig. 3(d), the accuracy for ID3 with SDQ outperforms the distributed SuLQ at all privacy budget levels.

VIII. Conclusion

In this work, we address the problem of distributed knowledge extraction while preserving privacy. We use differential privacy to limit the potential information exposure about individual records during the data mining process. However, direct use of differential privacy primitive SuLQ in privacy-preserving data mining is weak against collusion attack. We proposed SDQ, which combines secure multiparty computation, differential privacy and random select to provide a group query capability across multiple databases while significantly reducing the probablity of successful collusion attack. Through experimentation and analysis we have shown that decision tree induction with SDQ can achieve higher privacy than direct adoption of secure sum and better accuracy than direct adoption of SuLQ primitive. SDQ is particularly effective when data is sparsely located on large number of databases.

IX. Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-0716306, CNS-0831628, and CNS-0746977.

References

[1] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Rec.*, vol. 33, no. 1, pp. 50–57, 2004.

[2] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *JOURNAL OF CRYPTOLOGY*. Springer-Verlag, 2000, pp. 36–54.

[3] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations*, vol. 4, p. 2003, 2003.

[4] F. Emekci, O. D. Sahin, D. Agrawal, and A. El Abbadi, "Privacy preserving decision tree learning over multiple parties," *Data Knowl. Eng.*, vol. 63, no. 2, pp. 348–361, 2007.

[5] N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: a comparative study," *ACM Comput. Surv.*, vol. 21, no. 4, pp. 515–556, 1989.

[6] C. Dwork, "Differential privacy: a survey of results," in *TAMC'08*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1–19.

[7] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*. Springer, 2006, pp. 265–284.

[8] C. Dwork, "Differential privacy," in *ICALP*. Springer, 2006, pp. 1–12.

[9] S. Agrawal and J. R. Haritsa, "A framework for high-accuracy privacy-preserving mining," in *ICDE*, 2005, pp. 193–204.

[10] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.

[11] B. Yang, H. Nakagawa, I. Sato, and J. Sakuma, "Collusion-resistant privacy-preserving data mining," in *KDD '10*, 2010, pp. 483–492.

[12] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2008.

[13] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS '07*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 94–103.

[14] A. Blum, C. Dwork, F. Mcsherry, and K. Nissim, "Practical privacy: the sulq framework," in *In PODS 05*. ACM, 2005, pp. 128–138.

[15] A. Friedman and A. Schuster, "Data mining with differential privacy," in *KDD '10*. New York, NY, USA: ACM, 2010, pp. 493–502.

[16] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[17] E. Kiltz, "Unconditionally secure constant round multi-party computation for equality, comparison, bits and exponentiation," in *TCC*. Cryptology, 2005.

[18] T. Nishide and K. Ohta, "Multiparty computation for interval, equality, and comparison without bit-decomposition protocol," in *PKC'07*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 343–360.

[19] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *SIGMOD '09*. New York, NY, USA: ACM, 2009, pp. 19–30.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[21] D. N. A. Asuncion, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html