

Secure Computation Over Distributed Databases

CHUNHUA SU[†] and KOUICHI SAKURAI^{††}

Association rule mining over distributed databases is a very useful technique in data mining domain. In doing association rule mining in multi-client's dataset, there are some serious concerns on privacy and security. Such concerns may prevent the parties from directly sharing their data. Prior randomization approaches to this problem generally occur some tradeoff between privacy and accuracy. In this paper, we present a secure multi-party computation scheme which can do association rule mining of multi-client's dataset while strictly respecting their privacy with maintaining the accuracy of the results.

1. Introduction

As we know, data mining is an widely used technology which has emerged as a means of identifying patterns and trends from large quantities of data. Association rules are used in different fields, such as: Marketing, Targeted Advertising, Floor Planning, Inventory Control, Churning Management. By using this technology, we can extract some useful information from huge amounts of data. The algorithms generally assume that necessary information from all clients is gathered at one central site. Most data mining tools operate by gathering all data into a central site, then running an algorithm on those gathered data. However, in this real world, many organization and the individual will have concern about their own privacy, they may be reluctant to share their own data to go on data mining unless there is privacy preserving data mining technology to make sure their privacy won't be violated or mis-used by other parties.

For example, suppose some company want to do joint association rules mining over the individual database for the marketing analysis. However, it will violate both companies and customers' privacy. The company don't want its commercial privacy to be known by other companies. At the same time, the customers are also reluctant to share their private information without any reasonable technique to protect their privacy.

This paper proposes a scheme against the

problem of computing association rules within such a scenario by using the secure multi-party computation protocol. We assume homogeneous databases: All sites have the same schema, but each site has information on different entities. The goal is to produce association rules with every party's data transactions globally, while preventing the private information be known by other parties.

Association rules mining techniques are generally applied to databases of transactions where each transaction consists of a set of items. In such a framework the problem is to discover all associations and correlations among data items where the presence of one set of items in a transaction implies (with a certain degree of confidence) the presence of other items. Association rules are statements of the form $X_1, X_2, \dots, X_n \Rightarrow Y$, meaning that if we find all of X_1, X_2, \dots, X_n in the transactions, then we have a good chance of finding Y . The probability of finding Y for us to accept this rule is called the confidence of the rule. We normally would search only for rules that had confidence above a certain threshold.

We may also ask that the confidence be significantly higher than it would be if items were placed at random into baskets. In many (but not all) situations, we only care about association rules involving sets of items that appear frequently in baskets. For example, we cannot run a good marketing strategy involving items that no one buys anyway. Thus, much data mining starts with the assumption that we only care about sets of items with high support; they appear together in many transactions. We then find association rules only involving a high-support set of items. That is to say that $X_1, X_2, \dots, X_n \Rightarrow Y$ must appear in at least a certain percent of the transac-

[†] Kyushu University, Department of Computer Science and Communication Engineering, Email: su@itslab.csce.kyushu-u.ac.jp

^{††} Kyushu University, Department of Computer Science and Communication Engineering, Email: sakurai@csce.kyushu-u.ac.jp

tions, called the support threshold. How to do the global support threshold counting with respecting clients' privacy is a major problem in privacy-preserving rules mining.

In this paper, we will propose a secure multi-party computation based protocol to do the association rules mining. In this protocol, we assume that all the clients own a database and they want to do the distributed association rules mining while they don't trust in each other, the task of this protocol is computing the global mining result while preserve every clients' privacy. For reducing the individual client's computing complexity, we propose the two mediators to cooperate in the association rules mining protocol. We can show that our protocol works with less communication complexity and computational complexity.

2. Background and Related Works

2.1 Overview of A-Priori Algorithm

The major steps in association rule mining are: frequent itemset generation and rule derivation. For mining the association rules, there is a most frequently used algorithm call A-Priori Algorithm proposed in¹⁾. This algorithm proceeds levelwise, The A-Priori algorithm uses the downward closure property, to prune unnecessary branches for further consideration. It needs two parameters, *minSupp* and *minConf*. The *minSupp* is used for generating frequent itemsets and *minConf* is used for rule derivation.

Here, we give a brief review of how to compute the support and confidence: Let $I = \{i_1, i_2, \dots, i_n\}$ be an itemset. Let DB be set of transactions, where each transaction T is an itemset such that $T \subseteq I$. Given an itemset $X \subseteq I$, a transaction T contains X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ has support s in the transaction database DB if $s\%$ of transactions in DB contain $X \cup Y$. The association rule holds in the transaction database DB with confidence c if $c\%$ of transactions in DB that contain X also contains Y . An itemset X with k items called k -itemset. So we define as following:

$$\text{support}_{X \Rightarrow Y} = \frac{|T_{X \cup Y}|}{|DB|},$$

it means that the support is equal to the percentage of all transactions which contain both X and Y in the whole dataset. And then we

can get that:

$$\text{confident}_{X \Rightarrow Y} = \frac{\text{support}_{X \Rightarrow Y}}{\text{support}_X}$$

The problem of mining association rules is to find all rules whose support and confidence are higher than certain user specified minimum support and confidence.

Apriori algorithm is an influential algorithm for mining frequent itemsets for Boolean association rules. This algorithm contains a number of passes over the database. During pass k , the algorithm finds the set of frequent itemsets L_k of length k that satisfy the minimum support requirement. The algorithm terminates when L_k is empty. A pruning step eliminates any candidate, which has a smaller subset. The pseudo code for Apriori Algorithm is following:

C_k : candidate itemset of size k

L_k : frequent itemset of size k

L_1 = frequent items;

For ($k = 1$; $L_k \neq \text{null}$; $k++$) do begin

C_{k+1} = candidates generated from L_k ;

For each transaction t in database do

Increment the count of all candidates in

C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with minSupport

End

Return L_k ;

With the Apriori algorithm, only frequent itemsets satisfy minimum support threshold can be generated. With these frequent itemsets we can get the final association rules as the output of this algorithm.

2.2 Related Works

The most frequently used data randomization technique is the Random Data Perturbation (RDP) methodology. This methodology is that adding the random noise to confidential numerical attributes. In²⁾, the authors proposed a reconstruction procedure which is possible to accurately estimate the distribution of original data values from the perturbed data. In privacy preserving data mining techniques. It is possible to develop accurate approximation models while respecting users' privacy concerns. However, in the³⁾, Evfimievski et al pointed out that the privacy breach will occur in²⁾'s proposal and proposed a new randomization techniques to mine association rules from transactions consisting of categorical items where the data has been randomized to preserve privacy of individual transactions. A privacy breach is a situation when, for some clients, the disclosure of its randomized private information to the server re-

veals that a certain property of unrandomized private information holds with high probability. The methodology which was proposed in³⁾ is also an randomization methodology.

Generally speaking, the privacy of a randomization technique model is calculated on the basis of the original data distribution and the perturbation function. However, this definition is insufficient in protecting privacy. We can easily see that in the RDP domain, if we find that from the approximate reconstruction of the distribution of the perturbed value Y will occur with a probability interval, For example, assume that we infer a joint probability distribution function for Y and a certain parameter T . Using this joint distribution, we may learn that Y will fall into a probability interval with the knowledge of the distribution of T .

Furthermore, in⁴⁾ the authors developed a new attacking method using random matrix-based spectral filtering technique to attack the²⁾'s RDP based proposal. In this paper, it is showed that Randomization technique preserves very little privacy. Random noise can be represented in the form of random matrices and random matrices have some properties from which we can estimate the original data. Random objects have predictable structures in spectral domain.

Association rules mining is one of the most used tasks in data mining. Ultimately, in randomization methods, there is a trade off between accuracy of data mining results and data security. Hence, these methods may not be suitable for mining data in situations requiring both high accuracy and high security.

Because the data mining technology can be implemented in many practical methods, so there is no a universal solution for Privacy Preserving Data Mining. Nowadays, the methods used for the privacy preserving data mining can be classified into two general methodologies: One is the data randomization technique and the other is the cryptography based technique, especially secure multi-party computation (SMC).

Although randomization approaches work quite well for mining frequent itemsets and general classification problems. One problem with the above mentioned randomization technique is the tradeoff between privacy and accuracy of the data mining results. We also can see that in many cases, we will suffer from privacy breach when using the randomization-

based techniques. So can we do better with other methodologies? We find that using Secure Multiparty Computation techniques can make it better.

In Purdue University, there is a group which is working on this Secure Multiparty Computation-based data mining subject. Their paper^{5),6)} showed us how to perform the privacy-preserving association rules mining. However, their proposals are on-line protocol, it needs that all the participants should go online and act the complex computation protocols. And more,⁶⁾'s secure scalar product protocol was attack successfully by Bart Goethals et al⁷⁾. As we mentioned above, we have to find a new secure multi-party computing protocol to execute the association rules mining with both high accuracy and security.

2.3 Secure Multi-party Computation

In the setting of multi-party computation, sets of two or more parties with private inputs wish to jointly compute some (predetermined) function of their inputs. The computation should be such that the outputs received by the parties are correctly distributed, and furthermore, that the privacy of each party's input is preserved as much as possible. Now, we give a brief review of this area. In Secure Multi-party Computation, we always assume that *semi-honest model* exists. The details of this problem can be found in Goldreich⁸⁾. So we can go straightforward to review this .

(1)*Semi-honest Model*: A semi-honest party follows the rules of the protocol giving its correct input, but it is very curious and it only tries to deduce information on the inputs of the honest parties by inspecting all the information available to the corrupted parties. This is somewhat realistic in the real world because parties who want to mine data for their mutual benefit will follow the protocol to get correct results. Also, a protocol that is buried in large, complex software can not be easily altered, so we always believe that a semi-honest party will never cheat in the protocol's process.

At first we should give out some important definitions:

DEFINITION 1 (*k*-Party Functionality)

A *k*-party functionality is a functionality,

$$f : (0, 1^*)^k \rightarrow (0, 1^*)^k$$

that is computable in probabilistic polynomial time.

DEFINITION 2 (Security in The Semi-honest Model(Two Parties)) A proto-

col A_1, A_2 for f is secure for there are two polynomial time probabilistic simulators S_1 and S_2 such that, for all x_1 and x_2 , the view of A_1 in the interaction of $A_1(x_1)$ with $A_2(x_2)$ is computationally indistinguishable from $S_1(x_1, f_1(x_1, x_2))$. Similarly, the view of A_2 in the interaction of $A_1(x_1)$ with $A_2(x_2)$ is computationally indistinguishable from $S_2(x_2, f_2(x_1, x_2))$.

The definition can be very easily extended to the k -party, rather than having just k simulators. In k -party model, an adversary who views some subset of the k parties in the protocol doesn't learn anything useful.

There exists a vast body of literature on secure multiparty computation. The paper by Yao^{9), 10)} build foundations for general secure multi-party computation.

Conceptually, a secure multiparty computation for function f can be viewed as an implementation of a trusted party T , which, upon receipt of the input values x_1, \dots, x_n from parties P_1, \dots, P_n , respectively, produces the output value $y = f(x_1, \dots, x_n)$.

The above definition says that a computation is secure if the view of each party during the execution of the protocol can be effectively simulated by the input and the output of the party. This is not quite the same as saying that private information is protected.

3. Privacy-preserving Association Rules Mining

3.1 Problem Definition

There is some distributed clients which want to get the global result from their data transactions. They all have great concern about their privacy and also they want to get the accurate result to help their following decision. No client should be able to learn contents of a transaction of any other client. And we want to use some cryptographic toolkits to construct a secure multi-party computation protocol to perform this task.

3.2 Description of the toolkits and architecture

A transaction database can be seen also as a binary matrix where each row corresponds to a transaction, each column corresponds to an item, and there is one in the entry (i, j) if and only if the transaction i contains the item j . We assume that every client has consistent transaction sizes, since different itemset sizes may reveal some information about the customers'

shopping habits.

In our privacy-preserving association rules mining, we have to use the Commutative Encryption scheme, Nowadays, commutative encryption becomes an important tool that can be used in many privacy-preserving protocols. We call an encryption scheme is commutative if the following conditions hold: For any given feasible encryption keys such as $K_1, K_2, K_3, \dots, K_n$, we can do the encryption like this:

$$\begin{aligned} E_{K_1}(\dots E_{K_n}[M]\dots) &= E_{K_n}[\dots E_{K_1}[M]\dots], \\ \forall M_1, M_2, \text{ such that } M_1 &\neq M_2 \text{ and for any} \\ \text{given } k, \epsilon < \frac{1}{2^k}, \text{ there is:} & \\ Pr(E_{K_1}(\dots E_{K_n}(M_1)\dots)) &= (E_{K_1}(\dots E_{K_n}(M_2)\dots)) \\ < \epsilon & \end{aligned}$$

And with this condition then we can get the same decryption even though the decryption order is different, such as:

$$D_{K_1}[\dots D_{K_n}[E_{K_1}[E_{K_2}\dots[M]]]] = M$$

Most symmetric encryption scheme (such as DES and AES) are not commutative, some public key encryption schemes are commutative, These properties of commutative encryption can be used to do re-encryption and decrypt the cipher without fixed order.

In this paper, we use a suitable commutative public key cryptosystem ElGamal to encrypt the transaction. Suppose the client A has public key (p, g, x) and private key a with $x = g^a \text{ mod } p$, and Bob has public key (p, g, y) with private key b with $y = g^b \text{ mod } p$.

When encrypting the transaction T , the clients A should generate the a random number k_A , one use his public key (p, g, x) , to compute $r_A = g^{k_A} \text{ mod } p$ $c_A = (m * x^{k_A}) \text{ mod } p$, the client A will send r_A and c_A as the cipher. As the decryption, client A can use his secret key a to compute $T = c_A * (r_A^a)^{-1} \text{ mod } p$

Note that if we used the RSA public key cryptosystem in this way, the players would have to share the value of n - the calculations won't work unless everything is "modded" using the same modulus. However if the players know the value of n and their own public/secret RSA key pair (e, d) then they can compute the values of p and q and hence given the public key of another player, they can also compute the secret key of that player. This means that RSA with shared modulus is insecure and should not be used in our protocol.

In our protocol, we assume that there are two non-colluding parties, one is *Computation Center* and the other is *Transactions Mix Center*.

It seems not so secure by doing this, but in the real world, When people hire a lawyer, they assume that lawyers is not colluding with other related parties against them. Companies assume that their consultants do not collude with their competitors. It also works the same in our protocol scene.

So, in our protocol, we assume the existence of k untrusted, non-colluding sites.

Untrusted implies that none of these sites should be able to gain any useful information from any of the inputs of the local sites.

Non-colluding implies that none of these sites should collude with any other sites to obtain information beyond the protocol.

This means that every party who participate in the protocol and doesn't trust in each other will pass its input through the other parties, and each of these inputs are meaningless information by other parties. However, if any of the parties combined their data, they would gain some meaningful information from the combined data. For this reason, we require that the sites be non-colluding. This assumption is realistic. Each site combines the shares of the data it has received using a secure protocol to get the required data mining result.

We summarize the two components of the computing architecture as below:

Computation Center: This site can perform as a key center and generate public and private key for each client. The association rules mining computation will also be done in this site.

Non-Colluding Mix Center: This site will arrange the transaction items' ID order in the matrix so that we can get the final result of the association rules mining. It also can mix the client's re-encrypted transactions to keep the anonymity when those transactions are sent to the Computation Center.

Now, we want to use the encryption tools and construct our protocol with the help of the Computation Center and Transactions Mix Center as following:

A. Agreement on The Matrix ID Order

This step is very important, because the fixed the matrix ID can identify the items when to reveal the final mining result. That is to say, all the clients should follow the same matrix ID and express the transaction items in a certain order. So how to execute this matrix ID agreement secure is very important problem to be solved. We propose two

methods to do the ID order agreement. The first one is to use the cryptographic e-voting scheme to vote the item's ID order that include anonymity, fairness, and accountability, and the Non-Colluding Mix Center will act as a bulletin board. The other one is completely decide by the Non-Colluding Mix center. The former one will suffer from more communication complexity and computation complexity. The latter is enough for the privacy requirement because the clients will only send their encrypted transactions to the Non-Colluding Mix Center. Non-Colluding Mix Center won't get any useful information to recover the client's privacy.

B. Keys Generation and distribution

In Computation Center site, at first the site will do the key agreement with each client to possess the encryption key to distribute the client's public keys securely. Here we want to apply group key management protocol provides a security framework for creating cryptographic groups on a network. Group communication can benefit from IP multicast to achieve scalable exchange of messages. However, there is a challenge of effectively controlling access to the transmitted data. For example, when a new client want to join to the protocol or an old client want to quit the protocol, the group management is very important to solve the privacy problem under such situation. In this site two encryption system have to be applied, one is the commutative public key encryption (E_{pk}, D_{pk}) and the other can be a normal public key encryption system (E_{co-pk}, D_{co-sk}) without specific requirement of being commutative or not, while pk and sk denote public key and secret, respectively.

C. Mixing the Transactions

The goal during the privacy-preserving data mining process is to achieve results without revealing the identity of the individual users or any information that may result in identifying different people. We propose that the Non-Colluding Mix Center should do this task. In this step, all the clients will send their encrypted transactions matrix to Non-Colluding Mix Center, the site will re-encrypt all the transactions matrix using all the other clients' public key and then mix them. Finally, Non-Colluding Mix Center sends the all re-encrypted and mixed transactions matrix to the Computation Center to execute the association rules mining. After this procedure, the transaction's

anonymity will be attained.

3.3 Privacy-preserving Protocol for Association Rules Mining

In order to implement our secure computation protocol, we assume that there are n clients join in this protocol, our protocol does not need all the clients to be on-line simultaneously.

Protocol Description

(1) Assume that there are n clients want to do a joint association rules mining. Firstly, All the clients make a agreement on the transactions matrix ID order and express all of their transactions in the form of the matrix.

(2) The Computing Center generates key pairs for all the clients respectively. For transfer the encryption keys securely and keep the clients' privacy against Non-Colluding Mix Center, Computing Center have to share a common public key $co - pk$ with all the clients. The Computing Center generates will send all the public keys pk_1, \dots, pk_n to Non-Colluding Mix Center and send client's public keys pk_i to the responding i_{th} clients, respectively.

(3) The client i encrypt his transaction T_i with share key $k_{co} - pk$ to get $\hat{T}_i = E_{k_s}(T_i)$. Then client i use the commutative public key encryption scheme to re-encrypt his transactions \hat{T} with the public key $E_{pk_i}(\hat{T})$

(4) The client i sends his re-encrypted transactions to Non-Colluding Mix Center, Non-Colluding Mix Center do the re-encryption with all the public keys except client's public key pk_i .

(5) When Computing Center receive all the re-encrypted and mixed transactions, it will first, decrypt the all the re-encrypted transactions by computing

$$D_{K_1}[\dots D_{K_n}[E_{K_1}[\dots[E_{K_n}[\{\hat{T}_1, \dots, \hat{T}_n\}]]]] = \{\hat{T}_1, \dots, \hat{T}_n\},$$

after that it can attain the mixed itemset $\{\hat{T}_1, \dots, \hat{T}_n\}$ encrypted by the share public key pk_s .

(6) Finally, the Computing Center can use another public system (E_{co-pk}, D_{co-sk}) to decrypt the mixed itemset and gets: $\{T_1, \dots, T_n\}$

4. Security and Complexity Analysis

4.1 Security Analysis

To analyze the security of the entire protocol, we should first analyze the information and pass through and revealed to each participating site when executing the protocol.

Computing Center View:

During the execution of the protocol, Computing Center does not learn anything, because all it sees are values without ID, because the Computing Center does not possess the Matrix ID to identify the items. So Computing Center only know the final result of the association rule mining but it can not recover the original itemset related to the individual clients from the mixed values without matrix ID order information. Therefore nothing is revealed related to the individual privacy of clients.

Non-Colluding Mix Center View:

During the execution of the protocol, Non-Colluding Mix Center only learns the matrix ID order, or even don't learn the matrix ID order if we use the anonymity e-voting scheme with some secret sharing techniques. For this security analysis, we assume that Non-Colluding Mix Center can possess the matrix ID order and all the clients' public key. So he can perform the adaptive chosen-ciphertext attack against the encrypted transactions.

Furthermore, because in our protocol, the clients i encrypted their transactions using a share public keys $co - pk$ to encrypt the transactions before re-encrypt the transactions with its own public keys pk_i which is also shared with Non-Colluding Mix Center before sending them to Non-Colluding Mix Center.

Ideally, a public-key encryption scheme should be semantically secure against adaptive chosen-ciphertext attack. Informally, this means that an adversary can learn nothing about the plaintext corresponding to a given ciphertext c , even when the adversary is allowed to obtain the plaintext corresponding to ciphertexts (not equal to c) of its choice. We henceforth denote this security notion as IND-CCA (indistinguishability against adaptive chosen-ciphertext attacks). that is to say that our scheme's security, which is under the semi-honest model, relies on the encryption's strength.

Since neither the clients, Computing Center nor the Non-Colluding Mix Center receive any additional information beyond what they are

already supposed to know, the entire protocol is secure. Of course, in all of these analyses, we assume that no collusion occurs between these parties and these parties follow the protocols. This assumption is realistic in the sense, with some certain legal binding contracts this may be enforced.

4.2 Complexity Analysis

Communication complexity in distributed computing mainly focuses on the number or bits required to execute computing functions or tasks. In our protocol, each site express their transactions data into a matrix, so we can see it clearly that for each itemset of size j , $O(n * j)$ bits must be sent during the execution of our protocol. At the final step of our protocol, for the association rules mining, assumes that there are C_j candidate itemsets of size j , there will be $O(\sum_{i=0}^m C_i * ni)$ bit needed to be sent at most, where m denotes the size of largest itemset which can be found.

For analyzing the computation complexity, the number of bits in the output of the encryption of an itemset from Non-Colluding Mix Center, which we denote it as t , is the most important factor. Also, we can see that The computation complexity is so different between each individual site. Computing Center will bear the most computation complexity, it has to decrypt $n * t$ bit for nt times.

5. Conclusion and Future Works

The main contributions of this paper is to propose a general framework of secure multi-party computation for privacy preserving association rules mining. For that the randomization methodologies are not good enough to attain the high accuracy and protect clients' information from privacy breach and the malicious attack, we show that how association rule mining can be done in this framework and prove that is secure enough to keep the clients' privacy. We also show that our procotols works with less communication complexity and communication complexity compared to other related schemes. In the future research, we hope to improve the efficiency of this secure multi-party computing approach. We want to develop a family of problems and solutions to privacy-preserving data mining. In the future research, a common framework with more formal and reliable for privacy preservation will enable next generation data mining technology to make substantial advances in alleviating pri-

vacy concerns.

References

- 1) R. Agrawal, T. Imielinski, and A. N. Swami. "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 2003
- 2) Rakesh Agrawal, Ramakrishnan Srikant. "Privacy-Preserving Data Mining". ACM SIGMOD Int'l Conf. on Management of Data, Dallas, May 2000.
- 3) Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, Johannes Gehrke. "Privacy Preserving Mining of Association Rules". Knowledge discovery and data mining (KDD2002).
- 4) Hillol Kargupta, Souptik Datta, Qi Wang, Krishnamoorthy Sivakumar. "Random Data Perturbation Techniques and Privacy Preserving Data Mining". 2003 IEEE International Conference on Data Mining, 2003. Knowledge discovery and data mining (KDD2002).
- 5) M. Kantarcioglu and C. Clifton. "Privacy-preserving distributed mining of association rules on horizontally partitioned data". Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. 2002.
- 6) J.S. Vaidya and C. Clifton. "Privacy preserving association rule mining in vertically partitioned data". Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002.
- 7) Bart Goethals, Sven Laur, Helger Lipmaa and Taneli Mielikainen. "On Secure Scalar Product Computation for Privacy-Preserving Data Mining", The 7 th Annual International Conference on Information Security and Cryptology, 2004
- 8) O. Goldreich, "Secure multi-party computation", Sept. 1998, (working draft). Online Available: <http://www.wisdom.weizmann.ac.il/oded/pp.html>
- 9) A. Yao. "Protocols for secure computations". In Proc. 23rd IEEE Symposium on Foundations of Computer Science (FOCS '82). IEEE Computer Society, 1982.
- 10) A. Yao. "How to generate and exchange secrets". In Proc. 27th IEEE Symposium on Foundations of Computer Science (FOCS '86). IEEE Computer Society, 1986.