

Impact Factor 6.1



Journal of Cyber Security

ISSN:2096-1146

Scopus

DOI

Google Scholar



More Information

www.journalcybersecurity.com



Crossref



Google

Scholar

scopus

Secure Association Rule Mining on Vertically Partitioned Database

Ms. Nisha Gupta (Research Scholar)

Dr. Kalpana Sharma (Assistant Professor)

Computer Science & Engineering Department, Bhagwant University Ajmer
(Rajasthan), India

Abstract: In this paper context of Data mining entails the discovery of unexpected but reusable knowledge from large unorganized datasets. In recent years, organizations in different fields have been required to collaborate to create new value. However, data mining among and within organizations has raised privacy and confidentiality concerns.

In our proposal, parties can not contribute to something other than the number of records, as well as the candidate item set. This study focuses on the private-set intersection as a substitute of the scalar product and shows that this intersection enables organizations to execute ARM on vertically divided data, allowing elastic information contribution while preserving privacy without escalating communication and totalling costs. Besides we spotlight on the statement that the number of protocol rounds among parties can be reduced and present three use cases in which the proposed scheme works more effectively than the alive schemes..

Keywords- Privacy preserving, association rules mining, association rule hiding, frequent itemsets, private set intersection.

Introduction: The ubiquity of internet-of-things (IoT) devices and the people who use them has generated tremendous amounts of data Worldwide. Furthermore, cloud storage penetration and increases in network speed make it possible to store and distribute the data as they grow. Cisco foresees a massive increase in internet traffic, projecting 4.8 ZB per year by 2022 [2]. Data mining employs and exploits the discovery and management of reusable and perhaps unexpected knowledge from large unorganized datasets. Many algorithms have been designed for efficient and automatic analysis of this data so that users can accumulate, assimilate, interpret, and understand the knowledge obtained. ARM is one of the majority common data-mining algorithms.

ARM is used to streamline sales, optimize e-commerce advertisements, and mitigate software development obstacles, among many other applications. Generally, ARM has been used to

aggregate data into one location and then mine those data [3]_[7]. When data are merged and mined, confidential information can easily be accumulated. It becomes necessary to shield the isolation of such data by protecting them from unauthorized exploitation. A privacy-preserving association rule-mining (PPARM) process has thus been proposed. It performs data mining while upholding data security and privacy requirements. Therefore, PPARM has attracted widespread attention as a technology for data sanctuary and isolation protection. Several schemes have been proposed to implement mining in distributed data environments. These schemes have been broadly separated into those that use secure multi-party computation (SMC) and those that use cryptographic techniques. A two-party secure-computation scheme was proposed to directly execute a computation protocol with input from two parties without the help of a third party [8].

SMC extends secure two-party computation to any party, allowing them to execute desired computations without sharing input values. No communication apart from the protocol is required. Therefore, it is necessary to design the protocol appropriately according to its purpose. Cryptographic technique-based schemes use encryption (e.g., homomorphic), and, because mining is performed with encrypted data, users lacking the secret key cannot obtain any useful information.

Data privacy is protected, allowing data owners to delegate mining activities to third parties. Decryption is certainly required to obtain results, and SMC and cryptographic technique-based schemes have their strengths and weaknesses. Thus, we choose a scheme that suits our use case. Distributed data environments can follow many patterns. The two main types of distributions are horizontal and vertical. Horizontal distribution: With horizontal distribution, different parties collect various record sets to determine common attributes. Their databases are horizontally distributed so that the columns are the same, but the rows are different.

Data owners often outsource data storage and mining tasks in ARM. Data owners need to store data on a cloud server or request mining tasks from a third-party service provider because their expertise, resources, and storage are insufficient for the amount of data accumulated and computational resources required for mining. When delegating data storage or mining to a third party, data privacy is important due to various security threats.

However, even if data are encrypted, or an appropriate algorithm is used, confidential information and mining results may be leaked because the data are entrusted to a third party. In addition, cloud services have risks such as setting errors, unauthorized access, and API vulnerabilities. More than 70 million records were stolen or leaked in 2018 due to poorly configured Amazon Simple Storage Service (Amazon S3) buckets [36]. In 2019, UpGuard

reported that data were open due to a setting error of Amazon S3 [37]. To comply with industry standards, data owners also need to understand access control and where data are stored. However, it may take time to check the settings, or the service provider may not provide accurate settings when using a cloud service. An (ISC)² survey found that one in four organizations had experienced a cloud security incident in the preceding 12 months.

According to the survey, the biggest challenges for cloud security are data loss (64%) and data privacy (62%).

In particular, it is difficult for multiple companies to use cloud services together. The companies would not want to use a private cloud because of the additional costs of design and operation. Even when using a public cloud, it is necessary to use a provider with no stake in the data owners to reduce the risk of collusion between the data owner and the cloud service provider. Therefore, this study proposes a scheme in which ARM can be performed only by the data owners without outsourcing data storage and mining.

This study assumes that ARM among organizations exists in different industries, where each organization has data and shares their results. Because not every party should share their data, we consider SMC-based schemes.

Vertical data distribution is used because each party has different data. The proposed ARM scheme (i.e., VC02) uses a scalar product for information sharing among parties [16]. With VC02, parties can share the number of common records without unnecessary information exchange. Conversely, parties cannot share anything other than the number of common records. Therefore, we concentrate on the private-set intersection instead of the scalar product. This study shows that the private-set intersection enables the execution of ARM on vertically partitioned data without changing communication and computation costs.

The contributions of this study are as follows:

- 1- The parties can exchange information specified by the parties and execute the ARM without outsourcing the ARM.
- 2- Flexible information exchange, such as the number of common records and record elements, including item sets, and whether or not thresholds are exceeded, can be accomplished.
- 3- Information exchange uses a private-set intersection, and only information determined in advance is shared. Related Works. Table 1 summarizes the characteristics of related research, including the proposed scheme.

Schemes	Data Environment	Outsource	Support Task	Shared Information	Data Privacy	Mining Result Privacy
Proposed	Vertical distro ¹	No	FIM ²	Record elements ⁴ , Support value ⁵ , Threshold value ⁶	Yes	Unnecessary
GLP+13 [7]	Central	Yes	FIM	Not shared	Yes	Yes
LSC+18 [28]	Central	Yes	FIM, ARM ³	Not shared	Yes	Yes
WHV+14 [15]	Horizontal distro	Yes	FIM, ARM	Not shared	Yes	No
TI4 [11]	Horizontal distro	No	FIM	Support value	Yes	Unnecessary
CKM17 [13]	Horizontal distro	No	FIM	Support value	Yes	Unnecessary
DR18 [31]	Horizontal distro	No	FIM	Support value	Yes	Unnecessary
LLC+16 [34]	Vertical distro	Yes	FIM, ARM	Threshold value	Yes	Yes
BC17 [35]	Vertical distro	Yes	FIM	Not shared	Partial	No
VC02 [16]	Vertical distro	No	FIM	Support value	Yes	Unnecessary
DR19 [25]	Vertical distro	No	FIM	Support value	Yes	Unnecessary

TABLE 1. A comparison of main features in related PPARM schemes.

Classified by data environment and labeled Central, Horizontal distribution, and Vertical distribution. In a central environment, if a data owner does not outsource, privacy protection is not important because only the data owner knows the raw data and mining results. Therefore, in general, outsource is assumed for PPARM in the central environment.

In a distributed environment, sharing/merging data and mining tasks may be outsourced to a trusted third party or maybe processed among parties without outsourcing. ARM can be divided into the task of extracting frequent itemsets, calculating support, and confidence (see section II.A) followed by comparing them to thresholds for the generation of association rules. Table 1 shows the former task as frequent itemset mining (FIM) and the latter as ARM. Given the frequent itemsets and their support values, data owners can generate association rules, so PPARM generally focuses on FIM. A Scheme that can execute ARM without sharing the result of FIM has been proposed. If the target dataset is held by multiple data owners, the data owners may share data for mining. In ARM, to determine if a candidate itemset is a frequent itemset, the data owner needs to know whether the support value of the candidate itemset exceeds the threshold value. Therefore, the support values of the candidate itemsets are shared, or only the results of whether or not they exceed the threshold value are shared to keep the support value secret. Information sharing is unnecessary when the data owner does not perform mining; thus, schemes [15], [35] that implement ARM tasks without sharing information by the data owner have been proposed. The raw data information held by the data owners must be kept secret from others, including other data owners.

Most schemes consider data privacy, but the partial database contents are known to the data outsourcing destination in BC17 [35]. Furthermore, when data are outsourced and mined by someone other than the data owners, the data owners want to protect the privacy of the mining results. Because the mining results are the property of the data owners, leaking them would

impair profits. For example, when building a sales promotion plan from sales data mining, leakage of mining results to competitors can adversely affect sales. Some schemes require the cloud server to know the mining result, in which case, the privacy of the mining result is not protected.

Our proposed scheme can perform FIM without outsourcing in a vertically distributed environment. The major difference from other schemes is that data owners can share various information.

II. PRELIMINARIES

A. ARM

ARM was originally proposed to find relationships among items from supermarket transaction data. The ARM problem can be formally stated. Let J be a set of all items, and database D consists of a set of transactions over J . Let $T \subseteq D$, $t \in T$ be a transaction over J . The tid is the transaction identifier and is defined in D to make the transaction unique. I is a set of items from J . $I \subseteq J$ and $I \subseteq D$. When the probabilities of itemset X or Y in the transactions are 30% and 10%, respectively, the probability that both itemsets are included in the transaction is predicted to be 3%. If both itemsets have a 15% chance of being included in the transaction, X and Y will be related. However, because it is necessary to indicate whether there is a high probability of buying Y when buying X or a high probability of buying X when buying Y , the association rules have a direction. Let $X \rightarrow Y$ be an association rule with antecedent X and consequent Y . The support and confidence are used to evaluate the association rules. The support of rule $X \rightarrow Y$ is defined as the ratio of transactions including X and Y as a whole: $Support(X \rightarrow Y) = |D_X \cap Y| / |D|$, where $|D_X|$ indicates the number of transactions that satisfy condition X . The confidence of rule $X \rightarrow Y$ is defined as the value obtained by dividing the number of transactions including X and Y by the number of transactions including X . $Confidence(X \rightarrow Y) = |D_X \cap Y| / |D_X|$. Given database D and two threshold values, $minsupp$ and $minconf$, $Support(X \rightarrow Y) \geq minsupp$ and $Confidence(X \rightarrow Y) \geq minconf$ in D .

B. APRIORI ALGORITHM

ARM can be divided into two phases. The first phase finds frequent itemsets that exceed $minsupp$. The second phase finds itemsets that exceed $minconf$ from the frequent itemsets determined in the first phase. The number of rule candidates increases sharply with the increasing number of items in the database. An apriori algorithm is proposed to find frequent itemsets efficiently. It skips the calculation by using the feature that the support of an itemset is less than or equal to the support of the sub-itemset. The sets of transactions are treated as a database with n rows and m columns to execute the algorithm more efficiently.

Each row and column correspond to a transaction and an item, respectively. Each entry in the database is either 0 or 1, specifying the presence or absence of an item. If the i -th row and j -th column in the database correspond to transaction t_i and item I_j , respectively, then the j -th entry in row i , denoted by $t_i[j]$, indicates whether t_i contains I_j .

C. DISTRIBUTED ARM

This study considers a vertically distributed database. Database D is split vertically into two sets of columns. D includes all items I_{all} and is divided into DB1 with a column set of items I_1 to I_m and DB2 with a column set of items I_{m+1} to I_{all} . Table 2 shows an example of the divided database. Parties A and B manage DB1 and DB2, respectively, and cannot browse the contents of the other DBs.

Manager	Party A			Party B		
t_{id}	I_1	...	I_m	I_{m+1}	...	I_{all}
t_1	1	...	1	0	...	1
...	...	...	...	...	...	...
t_i	0	...	1	1	...	0
...	...	...	...	...	...	...
t_n	0	...	1	1	...	1

TABLE 2. Divided transaction database.

Algorithm for association rule mining of a vertically distributed database

```

1   $L_1 = \{\text{large 1-itemsets}\}$ 
2  for ( $k=2$ ;  $L_{k-1} \neq \emptyset$ ;  $k++$ ) do begin
3       $C_k = \text{apriori-gen}(L_{k-1})$ ;
4      for all candidates,  $c \in C_k$ , do begin
5          if all the items in  $c$  are entirely at A or B
6              that party independently calculates  $c.count$ 
7          else
8              let A have items  $a_1, \dots, a_p$  and B have items
                $b_1, \dots, b_q$ 
9              A calculates  $\vec{X}[i] = \prod_{j=1}^p t_i[I_{a_j}]$  for  $i = 1, \dots, n$ 
10             B calculates  $\vec{Y}[i] = \prod_{j=1}^q t_i[I_{b_j}]$  for  $i = 1, \dots, n$ 
11             compute  $c.count = |\vec{X} \cap \vec{Y}|$ 
12             endif
13              $L_k = L_k \cup c \mid c.count \geq \text{minsupp}$ 
14         end
15     end
16     Answer =  $\cup_k L_k$ 

```

$C_k = \text{apriori-gen}(L_{k-1})$: Generate k items candidate sets C_k from $k-1$ items frequent sets L_{k-1}

ALGO 1. This algorithm extends the apriori algorithm to the vertically distributed database.

necessary to calculate the number of transactions, including the candidate itemset. If the candidate itemset is composed of only A and B items, the entities mutually calculate the number of transactions. Conversely, if the candidate itemset exists across A and B, it is

necessary to exchange information with each other. First, each party generates an n -dimensional vector for each item in the candidate itemset. If the number of transactions is larger than $minsupp$, it is regarded as a frequent itemset. Repeat this process until there are no more candidate itemsets. Frequent and candidate itemsets need to be shared, but confidential information of each party is not disclosed in itemset sharing because the frequent itemsets are shared only in the final step.

D. SECURITY MODEL

This study assumes a semi-honest adversary because the use case entails communication among trusted organizations.

1) SEMI-HONEST ADVERSARIES

In this model, both parties follow the actions they are supposed to take according to the protocol. However, they can try to deduce more information from the data they obtain during execution. We follow [42] and assume that participants in our protocol include a receiver, R , and a sender, S . The receiver, R , executes the protocol and receives the final result. The sender, S , provides information per the protocol. The definitions of receiver and sender security are as follows:

2) THE SECURITY OF THE RECEIVER INDISTINGUISHABILITY

This security requires that the sender cannot distinguish between R and S , even if the inputs of the receiver are different.

3) THE SECURITY OF THE SENDER COMPARISON TO THE IDEAL MODEL

This security requires that the receiver cannot obtain more information than specified. An ideal implementation will formalize that definition. The ideal model implements a third party that obtains inputs from both parties and outputs the result. The security of the sender requires that the output of the protocol is indistinguishable from the ideal implementation.

III. PROPOSED SCHEME

A. OVERVIEW

The algorithm revealed in Figure 1 does not bring up how to estimate the number of function from the column vectors derived by each party.

This allowed the two parties to share only the number of transactions, as well as the candidate itemset. It is enough to simply extract the association rules. However, if the party wants to use rules other than those of association, the shared information may be insufficient. Therefore, it is desirable that the parties flexibly change the information to be shared according to their purposes. Thus, our scheme enables flexible information sharing via

the replacement of the scalar product from VC02 [16] with a private matching from FNP04 .

Our scheme has the following advantages:

- i In addition to the number of transactions that include the candidate itemset, parties can share which transactions include the candidate itemset and whether the number of transactions, including those of the candidate itemset, exceeds a threshold. Each party can choose what information they want to share.
- ii The information shared by each party is not known to the other party.
- iii Because the anticipated scheme can share the same information when using a scalar product, it is also possible to ensure that the same rules are generated in advance.

B. PRIVATE SET INTERSECTION

As mentioned in section II.C, each party needs to exchange information and calculate the required value on line 11 of Figure 1. In this study, we propose to implement line 11 using the private set intersection.

In other words, we replace line 11 with a protocol where the inputs are X and Y , generated on lines 9 and 10, and the output is $c:count$. Furthermore, private set intersection allows output other than $c:count$. In this paper, the protocol that outputs only the number of transactions, including the candidate itemset, such as scalar product computation, is called *private cardinality matching*. Similarly, the protocol that outputs the elements of the transactions, including the candidate itemset, for which the sum of the elements has the same result, is called *private matching*. Additionally, the protocol that outputs the result of whether the number of transactions, including the candidate itemset, exceeds a threshold is called *private matching for cardinality threshold*. We show the flow of the protocol in Figure 2.

Each party can change its shared information by changing the value sent during step iv of the Set Intersection phase. Because private matching is the basic pattern, private matching is described prior to the other two patterns. Inputs of protocols are EX and EY . However, "1" and "0" in the column vector is replaced with the corresponding *tid* and random value not included in *tid*, respectively. Protocol output has three patterns: (a) elements of transactions, (b) the number of transactions, and (c) whether the number of transactions is greater than the threshold value. We assume participants in our protocol are a receiver, R , and a sender, S .

1) PRIVATE MATCHING

This section presents the patterns sharing the most information. Here, the parties share elements of transactions, including the candidate itemset. First, R generates a polynomial, $P(x)$, whose root is the input EX . $D(x)$ is a polynomial of degree n :

$$\begin{aligned}
 P(y) &= (x_1 - y)(x_2 - y) \cdots (x_n - y) \\
 &= \alpha_0 + \alpha_1 y + \cdots + \alpha_n y^n \\
 &= \sum_{u=0}^n \alpha_u y^u.
 \end{aligned}$$

Protocol for private matching

INPUT: R 's input is a set $\bar{X}$, S 's input is a set $\bar{Y}$

OUTPUT: Pattern selected by participants from (a) to (c)

1. **Setup.** Let $\text{Enc}_{pk}(\cdot)$ be a semantically secure homomorphic encryption with a public key, pk . Let pk be shared with the parties. Parties should agree on what to share for transactions, including the candidate itemset (i.e., (a) elements of transactions, (b) the number of transactions, and (c) whether the number of transactions is greater than the threshold value).
2. **Set Intersection.** In this step, parties share information based on what parties agreed to during the **Setup** phase.
 - i. First, party R computes $P(y) = (x_1 - y)(x_2 - y) \cdots (x_n - y) = \alpha_0 + \alpha_1 y + \cdots + \alpha_n y^n = \sum_{u=0}^n \alpha_u y^u$, and encrypts the coefficients to $\{\text{Enc}_{pk}(\alpha_0), \dots, \text{Enc}_{pk}(\alpha_n)\}$.
 - ii. R sends $\{\text{Enc}_{pk}(\alpha_0), \dots, \text{Enc}_{pk}(\alpha_n)\}$ to S .
 - iii. For $y \in Y$, S computes $\text{Enc}_{pk}(P(y))$, in particular, $\text{Enc}_{pk}(\alpha_0) \cdot y^0 + \text{Enc}_{pk}(\alpha_1) \cdot y^1 + \cdots + \text{Enc}_{pk}(\alpha_n) \cdot y^n = \text{Enc}_{pk}(\alpha_0 \cdot y^0) + \text{Enc}_{pk}(\alpha_1 \cdot y^1) + \cdots + \text{Enc}_{pk}(\alpha_n \cdot y^n) = \text{Enc}_{pk}(\sum_{u=0}^n \alpha_u y^u) = \text{Enc}_{pk}(P(y))$, by using the homomorphic properties.
 - iv. S chooses a random value, r , and computes $\text{Enc}_{pk}(rP(y) + z)$. z is the following value based on what to share: (a) $z = y$, (b) $z = **$, where $**$ presets some unique strings, (c) $z = r_y$, for random r_y .
 - v. S sends n randomly sorted ciphertexts to R .
 - vi. R decrypts all n ciphertexts received. R outputs values $x \in X$ for which there is a corresponding decrypted value.

Flexible information sharing between R and S without revealing information. R expands the polynomial, encrypts the coefficients using homomorphic encryption, and sends $\text{fEnc}_{pk}(_0); \text{Enc}_{pk}(_n)g$ to S . S generates a ciphertext of $P.y/$ using homomorphic properties. Then, S

generates a value to send to R by using random number r for all elements y of EY , as follows: $Encpk .P .y/ _ Encpk .r/ C Encpk .y/ D Encpk .rP .y/ C y/$, and $P .y/ D 0$ when y is included in the input of R . Thus, the calculation result of S is $Encpk .y/$. On the other hand, it will be a random value when y is not included in the input of R . Because R can decrypt the ciphertext, R decrypts all values sent from S , and obtains the tid , including the candidate itemset. Finally, R shares its results with S .

2) PRIVATE CARDINALITY MATCHING

This protocol pattern limits the information to be shared. Protocols do not share elements of transactions. Instead, they share only the number of transactions. In other words, this pattern shares the same information as the scalar product used in VC02. We can implement this pattern with only a small change to the behaviour of private matching. In the step where S calculates ciphertext, S enters a unique string, instead of y . R decrypts the ciphertext received from S and counts the number of cipher texts from which was obtained. The total number of this ciphertext matches the number of transactions, including the candidate itemset.

IV. ANALYSIS

A. SECURITY ANALYSIS

The output of the algorithm in Figure 1 includes itemsets having k elements that are candidates for frequent itemset. If $L1$ is to be shared with the other party, they exchange information only on line 11. Therefore, the security of the anticipated scheme depends on the data-sharing protocol. The proposed scheme uses FNP04 as the information-sharing protocol, which ensures the security claims in the semi-honest model as follows:

Lemma 1 (Correctness): The protocol participants can obtain the desired result only if R and S share a common value by calculating according to the protocol for private matching. In particular, R can get the encrypted y during step vi during the Set Intersection phase if $x D y$. Otherwise, R obtains the encrypted random number.

Lemma 2 (security of R is preserved): Based on the input of R to the protocol, R calculates the polynomial $P .y/$ and sends the information to S . Specially, R calculates $P .y/$ by using the input value as a root. It encrypts the coefficient of the $P .y/$ and sends S the encrypted value. Because the information obtained by S is only the encrypted coefficient, S cannot distinguish the input of R . Thus, when the encryption scheme is secure, the privacy of R is protected.

Lemma 3 (security of S is preserved): The ideal model assumes a third party who takes the input EX of R and the input EY of S and outputs the result of protocols. In this case, the

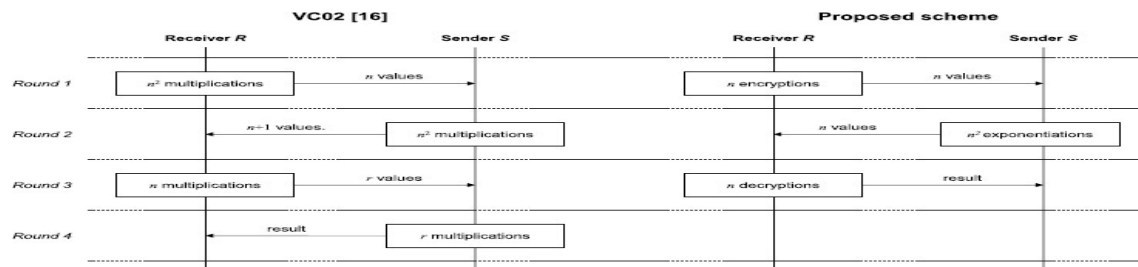


FIGURE 3. Protocols for sharing information for VC02 and the proposed scheme are illustrated. The content of each box shows the computational overhead of each round. The label on each arrow shows the message to be transmitted. information that R can obtain from the third party is only the result of $X \setminus Y$. In the real model, R can only obtain $\text{Enc}(y)$ or $\text{Enc}(r)$, where r is a random value, so the information that

can be obtained is indistinguishable from that of the ideal model; thus, the privacy of S is protected.

B. COMMUNICATION/COMPUTATION ANALYSIS

Table 3 shows the results of a comparison between VC02 and the proposed scheme. The key difference is the type of information that can be shared among parties. VC02 can share only one type of information, but the proposed scheme can share three types. There is no difference in the amount of communication per round, but there is a difference in the number of communication rounds among parties. VC02 requires four rounds of communication until the results are obtained, whereas the proposed scheme requires three rounds. Figure 3 shows the flows of protocols for information sharing. In VC02, two parties, simply referred to as A and B , execute a four-round protocol on line 11 of Figure 1 to perform scalar product. In this study, for the sake of consistency, the party that starts the protocol is called a receiver R , and the other party is called a sender S . First, R sends a message having n values to S , and S responds with a message composed of $n-1$ values. In response, R sends a message consisting of r values, where r is a random number determined by S and satisfies $n > r$. Finally, S calculates the final result and sends it to R . Communication overhead is proportional to n and can be expressed as $O(n)$. In the protocol of the proposed scheme, n ciphertexts and the final result are transmitted for three rounds. Communication cost overhead of the proposed scheme is $O(n)$ as with VC02. In VC02, the calculation for generating n values that R and S send first requires the largest calculation cost

	VC02 [16]	Proposed scheme
Shared Information among parties	Number of transactions, including the candidate itemset.	- Elements of transactions, including the candidate itemset. - Number of transactions, including the candidate itemset. - Whether the number of transactions, including the candidate itemset, is greater than the threshold value.
Communication cost	$O(n)$ Four rounds	$O(n)$ Three rounds
Computation cost	$O(n^2)$	$O(n^2)$

n : the number of items to communicate.

TABLE 3. Comparison of schemes.

Because n multiplications are needed to make one value, a total of n^2 multiplications are required. Thus, the overhead of the calculation cost is $O(n^2)$. In the proposed scheme, R computes the coefficients of the polynomial using interpolation and performs n instances of encryption and decryption. S has the largest computation overhead, and the processing of step iii during the Set Intersection phase requires a computation overhead of $O(n^2)$ exponentiations. In this round, the computation overhead of $Enc_{pk} .P .y//$ is $O(n)$ exponentiations because it is indispensable to compute yn . In addition, n multiplications of homomorphically encrypted values are required. Because these multiplications are actually implemented as exponentiations, the total overhead is $O(n^2)$ exponentiations. Freedman *et al.* [46] mention reduction of the computational overhead and explain that using a hash function reduces the overhead to $O(n \ln \ln n)$. However, the reduction of the computation cost is outside the scope of this study because the proposed scheme has features other than the computation cost.

V. USE CASES

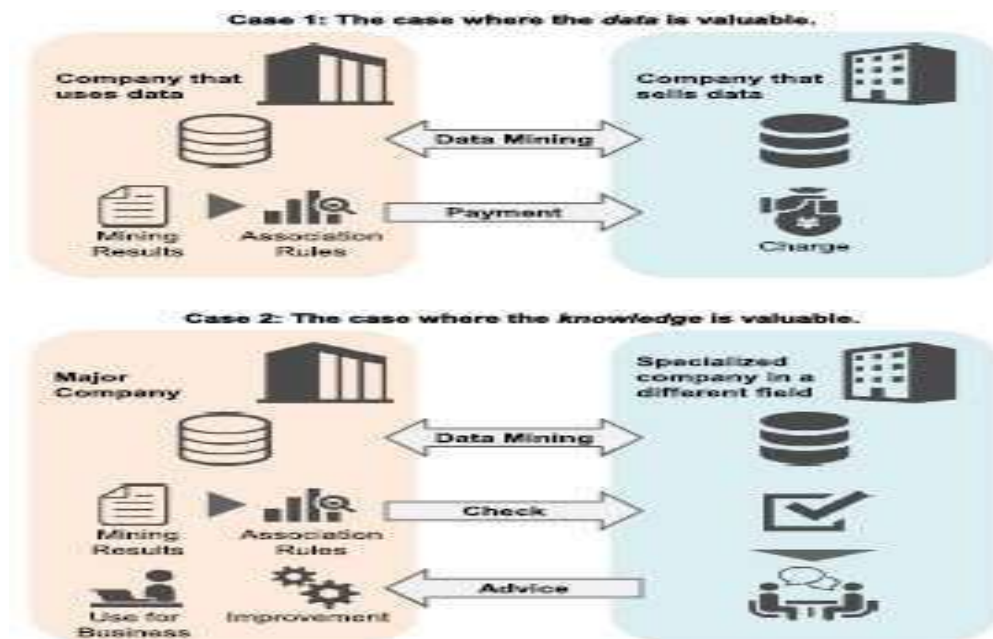
The major difference between protocols of the proposed scheme and VC02 is the number of rounds. Due to the difference, R and S execute the final round in the proposed scheme and VC02, respectively. In the final round, the party shares the final result of the protocol with the other in both schemes. Therefore, if the recipient of the final round does not require the final result, the final round can be omitted. The recipient of the final round is S in the proposed scheme and R in VC02. Because R starts the protocol, R should need the results of the protocol. On the other hand, we believe that there are cases where R does not require the results. In these cases, R expects another benefit by cooperating with the protocol and providing its own data. We present three use cases for it. We organize use cases based on the concept of the data information knowledge wisdom (DIKW) hierarchy.

This is a framework for interpreting information and consists of four layers: *Data*, *Information*, *Knowledge*, and *Wisdom*. The four layers are defined as follows according to Data are defined as symbols that represent properties of objects, events, and their environment. They are of no use until they are in a useable form.

If we replace raw data, such as sales data, with "*data*," frequent itemsets with "*information*," and association rules with "*knowledge*," we can consider ARM with this concept. Based on this concept, the proposed scheme and VC02 are schemes for safely sharing *information*. However, in a case where the emphasis is on something other than *information*, sharing of *information* is unnecessary, and the protocol could omit the final round. Therefore, Figure 4 shows each use case that has value in *data*, *knowledge*, and *wisdom*.

Case 1: The case where the *data* are valuable. If a company has valuable data, it will receive a payment for providing the data and cooperating with the mining. The company may be specialized in collecting data and selling data to the companies that need data. It may be unnecessary to share the final result of the protocol when demanding remuneration other than information for providing data.

Case 2: The case where the *knowledge* is valuable. The final result of the protocol in the proposed scheme and VC02 is frequent itemsets. Parties know which items are likely to appear in the same data from the frequent itemsets but do not know which items are appropriate as an antecedent or a consequent. Parties cannot properly use the



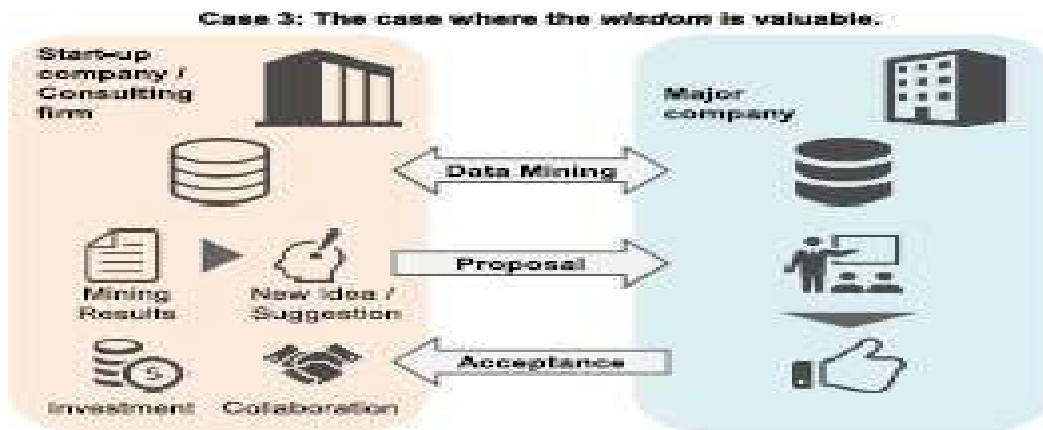


FIGURE 4. Use cases that have value in data, knowledge, and wisdom. mining results until they understand the relationship between the antecedent and the consequent. We consider the concept of open innovation. Suppose that a major company asks for cooperation from a specialized company in a different field. The major company wants to develop new products or services by combining its data and specialized data of the specialized company. However, the major company cannot understand the details of the data because the data of the specialized company is from a different field. The major company asks the specialized company for advice. Association rules may help the specialized company give appropriate advice because relationships among items are more intuitive. A form, such as knowledge, is preferred for these use cases because that form is easy for people to understand.

Case 3: The case where the *wisdom* is valuable. We consider the case of combining the data of a major company and the data of a startup company to create a new business. The major company wants services and solutions based on new ideas the startup company has derived from mining results. In other words, the major company requires the startup company to propose their own wisdom, not information or knowledge. When the major company is impressed with the wisdom, it offers funding and collaboration opportunities to the startup company.

In addition, we could make this case by replacing the startup company with a consulting firm. Mining is carried out using the unique data held by each of them as input. The consulting firm sublimates the information and knowledge obtained from mining into wisdom and proposes business improvements and management strategies to the client company.

In other words, the client company pays the consulting fee for the wisdom proposed by the consulting company. These use cases require unique and personal insights rather than the

information and knowledge required for mining. Therefore, the proposed scheme is more appropriate than VC02 for the aforementioned use cases.

VI. CONCLUSION AND FUTURE WORK

This paper presented a secure ARM scheme for vertically partitioned data. The proposed scheme enables flexible information sharing by using private-set intersections. Compared with existing ARM schemes that use scalar products, our communication and calculation costs were comparable, and multiple information sharing patterns were achieved. Furthermore, the number of protocol rounds could be reduced from the existing scheme. Focusing on this point, we presented use cases for which the proposed scheme works effectively.

In the future, we plan to perform stricter security analyses on the wide use of privacy-preserving data-mining techniques. Furthermore, we plan to investigate other algorithms and consider more efficient data-mining techniques.

REFERENCES

- [1] D. Reinsel, J. Gantz, and J. Rydning. *The Digitization of the World From Edge to Core*. IDC. Accessed: Sep. 30, 2019. [Online]. Available: https://www.seagate.com/_les/www-content/our-story/trends/_les/idcseagate-dataage-whitepaper.pdf
- [2] C. Public. *Cisco Visual Networking Index: Forecast and Trends, 2017-2022*. Cisco. Accessed: Sep. 30, 2019. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf>
- [3] W. K. Wong, D. W. Cheung, B. Kao, N. Mamoulis, and E. Hung, "Security in outsourcing of association rule mining," in *Proc. 33rd Int. Conf. Very Large Data Bases (VLDB)*, Vienna, Austria, 2007, pp. 111-122.
- [4] I. Molloy, N. Li, and T. Li, "On the (In)Security and (Im)Practicality of outsourcing precise association rule mining," in *Proc. 9th IEEE Int. Conf. Data Mining*, Washington, DC, USA, Dec. 2009, pp. 872-877.
- [5] C.-H. Tai, P. S. Yu, and M.-S. Chen, "K-support anonymity based on pseudo taxonomy for outsourcing of frequent itemset mining," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Washington, DC, USA, 2010, pp. 473-482.
- [6] F. Giannotti, L. V. S. Lakshmanan, D. Pedreschi, H. Wang, and A. Monreale, "Privacy-preserving data mining from outsourced databases," in *Computers, Privacy and Data Protection: an Element of Choice*, S. Gutwirth, Y. Poullet, P. De Hert, R. Leenes, Eds. Dordrecht, The Netherlands: Springer, vol. 2011, pp. 411-426.

- [7] F. Giannotti, L. V. S. Lakshmanan, A. Monreale, D. Pedreschi, and H. Wang, "Privacy-preserving mining of association rules from outsourced transaction databases," *IEEE Syst. J.*, vol. 7, no. 3, pp. 385_395, Sep. 2013.
- [8] A. C. Yao, "Theory and application of trapdoor functions," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (sfcs)*, Nov. 1982, pp. 80_91.
- [9] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1026_1037, Sep. 2004.
- [10] A. Schuster, R. Wolff, and B. Gilburd, "Privacy-preserving association rule mining in large-scale distributed systems," in *Proc. IEEE Int. Symp. Cluster Comput. Grid*, Chicago, IL, USA, Apr. 2004, pp. 411_418.
- [11] T. Tassa, "Secure mining of association rules in horizontally distributed databases," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 970_983, Apr. 2014.
- [12] X. Juan and Z. Yanqin, "Application of distributed oblivious transfer protocol in association rule mining," in *Proc. 2nd Int. Conf. Comput. Eng. Appl.*, Washington, DC, USA, 2010, pp. 204_207.
- [13] H. Chahar, B. N. Keshavamurthy, and C. Modi, "Privacy-preserving distributed mining of association rules using Elliptic-curve cryptosystem and Shamir's secret sharing scheme," *Sadhana*, vol. 42, no. 12, pp. 1997_2007, Dec. 2017.
- [14] C. N. Modi and A. R. Patil, "Privacy preserving association rule mining in horizontally partitioned databases without involving trusted third party (TTP)," in *Proc. 3rd Int. Conf. Adv. Comput., Netw. Inform. (ICACNI)*. New Delhi, India: Springer, 2015, pp. 549_555.
- [15] O. A. Wahab, M. O. Hachami, M. Vivas, G. G. Dagher, and A. Zaffari, "DARM: A privacy-preserving approach for distributed association rules mining on horizontally-partitioned data," in *Proc. 18th Int. Database Eng. Appl. Symp.*, Porto, Portugal, 2014, pp. 1_8.
- [16] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Edmonton, AB, Canada, 2002, pp. 639_644.
- [17] S. Zhong, "Privacy-preserving algorithms for distributed mining of frequent itemsets," *Inf. Sci.*, vol. 177, no. 2, pp. 490_503, Jan. 2007.
- [18] J. Vaidya and C. Clifton, "Secure set intersection cardinality with application to association rule mining," *J. Comput. Secur.*, vol. 13, no. 4, pp. 593_622, Oct. 2005.
- [19] X. Ge, L. Yan, W. Shi, and J. Zhu, "Privacy-preserving distributed association

rule mining based on the secret sharing technique," in *Proc.2nd Int. Conf. Softw. Eng. Data Mining*, Chengdu, China, Jun. 2010,pp. 345_350.

[20] R. Kharat, M. Kumbhar, and P. Bhamre, Eds., ``Ef_cient privacy preservingdistributed association rule mining protocol based on randomnumber," in *Proc. Intell. Comput., Netw., Inform.*, New Delhi, India, 2014,pp. 827_836.

[21] B. Rozenberg and E. Gudes, ``Association rules mining in verticallypartitioned databases," *Data Knowl. Eng.*, vol. 59, no. 2, pp. 378_396,Nov. 2006.

[22] H. Hammami, H. Brahmi, S. Ben Yahia, and I. Brahmi, ``Using homomorphicencryption to compute privacy preserving data mining in a cloudcomputing environment," in *Proc. Eur., Medit., Middle Eastern Conf. Inf.Syst. (EMCIS)*, Coimbra, Portugal, 2017, pp. 397_413.

[23] M. Waddey, P. Poncelet, and S. Ben Yahia, ``A novel approach for privacymining of generic basic association rules," in *Proc. ACM 1st Int.WorkshopPrivacy Anonymity Very Large Databases (PAVLAD)*, Hong Kong, 2009,pp. 45_52.

[24] B. Wang, Y. Zhan, and Z. Zhang, ``Cryptanalysis of a symmetric fullyhomomorphic encryption scheme," *IEEE Trans. Inf. Forensics Security*,vol. 13, no. 6, pp. 1460_1467, Jun. 2018.

[25] N. Domadiya and U. P. Rao, ``Privacy preserving distributed associationrule mining approach on vertically partitioned healthcare data," in *Proc.Procedia Comput. Sci.*, Fez, Morocco, 2019, pp. 303_312.