Innovation, Technology and Management Journal
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

STATE OF THE ART OF SECURE MULTIPARTY COMPUTATION FOR PRIVACY PRESERVING DATA MINING

ESTADO DA ARTE DE *SECURE MULTIPARTY COMPUTATION* PARA MINERAÇÃO DE DADOS COM PRESERVAÇÃO DE PRIVACIDADE

Walter Priesnitz Filho[1] and Carlos Ribeiro[2]
[1]Programa Doutoral em Segurança de Informação
Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
walter.filho@tecnico.ulisboa.pt
[2]Programa Doutoral em Segurança de Informação
Instituto Superior Técnico, Universidade de Lisboa Lisboa, Portugal
carlos.ribeiro@ist.utl.pt,

Abstract
In this paper we present the State of the Art in Secure Multiparty Computation Protocols to Privacy Preserving Data Mining. Basic definitions on the main topics are presented as some proposed protocols in the reviewed literature. We also briefly discuss the authors contribution under practical integration perspective and points some issues on using SMC in PPDM.
Keywords: Secure Multiparty Computation, Privacy-preserving Data Minig, Privacy, Information Security

Resumo
Neste artigo apresentamos o estado da arte dos protocolos de Secure Multiparty Computation para mineração de dados com preservação de privacidade.São apresentadas as definições básicas dos principais tópicos dos protocolos propostos da literatura estudada. Também é apresentada uma breve discussão sobre a contribuição dos autores sob um perspectiva prática de integração e apontados alguns aspectos a tomar cuidado quando da utilização da SMC em PPDM.
Palavras-chave: Secure Multiparty Computation, Mineração de Dados com Preservação de Privacidade, Privacidade, Segurança de Informação

D.O.I.: 10.7198/geintec.v7.i4.1213

1 Introduction

Recent advances in information technology enable more organizations to collect, store, and use several types of user's/clients' information, Tassa and Gudes (2012). Such large repositories of data carry valuable information that may be extracted through data mining tools. In such settings, protecting the privacy of the user's/clients' private data in those repositories is very important. Identifying attributes such as names and ID numbers use to be removed before releasing the data sets for mining purposes, but the private information might still leak due to linking attacks. Such attacks may join the public attributes, also known as quasi-identifiers, from published data with a publicly accessible table like voters registry, and thus disclose private information of specific individuals.

Several organisations like Business Corporations, Government Departments, Research, profit and non-profit, attain and analyse data to fulfil their desired goals. The data utilised usually embodies classified information of private entities. Murugeshwari, Jayakumar, and Sarukesi (2012)

Privacy-preserving data mining, according to Lindell and Pinkas (2009), considers the problem of running data mining algorithms on private data that is not supposed to be revealed — even to the party/entity running the algorithm. Considering a distributed set of parties, the data is divided among two, or more, different parties and run a data mining algorithm on the union of the parties' databases without allowing any party to view another individual's private data.

With Secure Multiparty Computation, from now on SMC, Mishra, Trivedi, and Shukla (2009), several parties can jointly perform some global computation using their private data without loss of data security/privacy. SMC provides a base for end-to-end secure multiparty protocol development.

This paper is strutured as follows: Section 2 presents an overview about Privacy Preserving Data Mining, in Section 3 we show some basic concepts about Secure Multiparty Computation, Section 4 presents some Secure Multiparty Computation techniques and protocols for PPDM, Section 5 bring some considerations about PPDM and SMC.

2 Privacy Preserving Data Mining

Privacy preserving data mining (PPDM) technique is a research area in data mining where mining algorithms are analysed for the side effect in data privacy Nivetha.P.R and selvi.K (2013). The objective of privacy preserving data mining is built algorithms for transforming the original information, in some way, keeping the private data and private knowledge confidential even after the mining process.

PPDM is defined, Ying-hua, Bing-ru, Dan-yang, and Nan (2011), as using accurate models and analysis results without access to the original data. Data privacy preserving technology can be achieved through data perturbation, secure multiparty computation and restricted queries.

Wang Wang (2010), adds stating that the main objective of PPDM is to develop data mining methods without increasing the risk of mishandling the data used. These techniques use some form of modification to the original data to accomplish the privacy preservation. The modified dataset is achievable for mining and must meet privacy requirements without missing the profit of mining.

Some of these techniques are described below:

2.1 Data Perturbation

Perturbation techniques work hiding part of the original data, and data-miners acts in the disturbed data, Ying-hua, Bing-ru, Dan-yang, and Nan (2011). The disturbed data keeps the properties the same as the original, so the knowledge is accurate even in the disturbed data. Common perturbation technologies are: Add Noise and Random Response.

2.2 Secure Multiparty Computation

Secure Multiparty Computation (SMC) provides a solution that can effectively protect the sensitive data. SMC considers a set of collaborators/parties who wish to mine their data collectively but does not want to disclose their datasets to each other, Gkoulalas-Divanis and Verykios (2009). In this way, this distributed PPDM problem can be reduced to the secure computation of a function with distributed inputs and solved using cryptographic approaches. Each party, Vaidya and Clifton (2004), knows/possess some of the private data join in a protocol that produces the data mining results, yet that can be demonstrated not to reveal data to parties that don't already had them. Thus the process of data mining does not cause breaches of privacy.

2.3 Restricted Queries

Anonymous data are used in a restricted query Ying-hua, Bing-ru, Dan-yang, and Nan (2011) it avoids those who can try to reconstruct the original data from a query. Anonymous techniques divide the properties of the original real data into four categories:

1. Individually Identifier Attribute used to only identify the body (employee number, ID number, and so on);

2. Quasi Identifier Attribute is a set of attributes which could identify the body based on background knowledge (birthday, gender, ZIP-Code);

3. Sensitive attributes that contain sensitive information (basic pay and allowances);

4. Not Sensitive Attribute is attributes that a Data-Miner not interested in.

Anonymous technology differs from the method of adding noise; it publishes sensitive information selectively ensuring the data cannot mine the identity of the data supplier. Some anonymous techniques:

2.3.1 *k*-Anonymization

K-Anonymization Tassa and Gudes (2012) generalises or suppresses the values of the public attributes when projected on the subset of public attributes, thus hiding its relationship with the values of the sensitive attribute. This generalisation, or suppression, occurs in such a way that each of the released records becomes indistinguishable from at least *k-1* other records. As a consequence, each individual may be linked to sets of records of size at least *k* in the released anonymized table, whence privacy is protected to some extent.

2.3.2 *l*-Diversity

l-Diversity is a method proposed by Machanavajjhala, Kifer, Gehrke, and Venkitasubramaniam (2007) to solve the problem of background knowledge attack and homogeneity attacks in *k*-anonymity. Each data set of size at least *k* of indistinguishable records must have at least *l* ”well-represented” distinct values in the sensitive attribute, Tassa and Gudes (2012). One of the ways how *l*-diversity is usually enforced is by demanding that the frequency of each of the private values within each data set of indistinguishable records does not exceed $1/l$.

2.3.3 *t*-Closeness

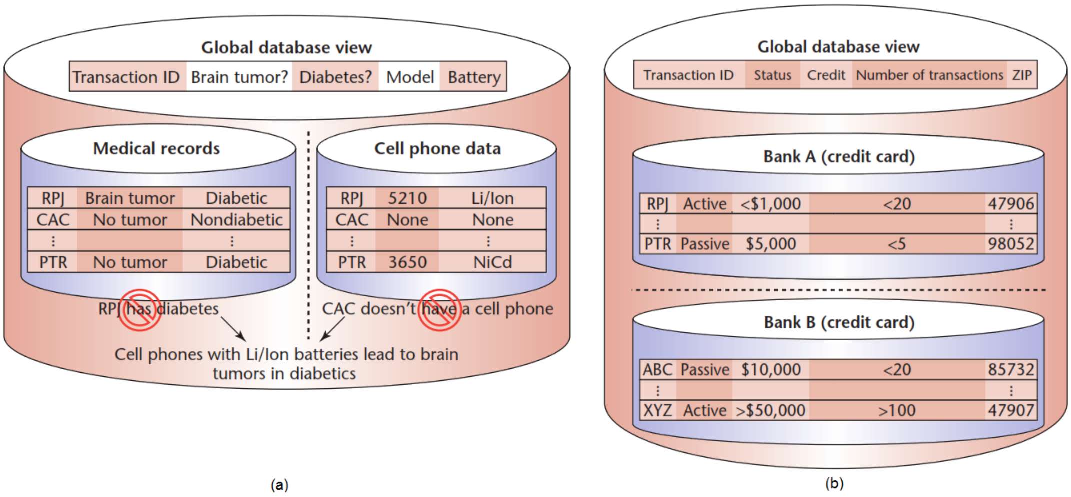
To avoid the problem with *l*-diversity, Li, Li, and Venkatasubramanian (2007) proposed *t*-Closeness method based on *k*-anonymisation and *l*-diversity method, which integrated the distribution of sensitive attributes. The authors stated that ”an equivalence class is said to have *t*-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold *t*. A table is said to have *t*-closeness if all equivalence classes have *t*-closeness”.

2.4 Data Partitioning

Considering, Vaidya and Clifton (2004), different parties trying to perform a PPDM and assuming that the input to a function is distributed among different sources, though, the privacy of each data source comes into question. The way the data is distributed also plays an important role in defining the problem because data can be partitioned into many parts either vertically or horizontally.

Vertical data partitioning implies that although different sites gather information about the same set of entities, they collect different feature sets. Moreover, in horizontal partitioning, different sites collect the same set of information but about different entities.

Figure 1: Data Partitioning: (a) Vertical data partitioning and (b) Horizontal data partitioning from Vaidya and Clifton (2004).


Global database view
Transaction ID | Brain tumor? | Diabetes? | Model | Battery
Medical records
RPJ | Brain tumor | Diabetic
CAC | No tumor | Nondiabetic
⋮ | | ⋮
PTR | No tumor | Diabetic
Cell phone data
RPJ | 5210 | Li/Ion
CAC | None | None
⋮ | | ⋮
PTR | 3650 | NiCd
RPJ has diabetes
CAC doesn't have a cell phone
Cell phones with Li/Ion batteries lead to brain tumors in diabetics
(a)
Global database view
Transaction ID | Status | Credit | Number of transactions | ZIP
Bank A (credit card)
RPJ | Active | <$1,000 | <20 | 47906
⋮ | | | | ⋮
PTR | Passive | $5,000 | <5 | 98052
Bank B (credit card)
ABC | Passive | $10,000 | <20 | 85732
⋮ | | | | ⋮
XYZ | Active | >$50,000 | >100 | 47907
(b)

2.5 Distributed Data Methods

Liu, Ying-hua, Bing-ru, Dan-yang, and Nan (2011), states that there are three kinds of methods of distributed data: association rules, clustering and classification.

2.5.1 Association Rules

Association rule learning in PPDM is a popular and well-researched method for discovering interesting relations between variables in huge databases. It shows attribute value conditions that frequently occur together in given databases.

2.5.2 Clustering

In Clustering Tassa and Gudes (2012), data is generalised according to accepted generalisation rules. Practitioners still perceive clustering-based privacy models as sufficient for mitigating risk in the real world while maximising utility, and real-life applications still utilise them for sanitising data.

2.5.3 Classification

A classification algorithm is used to classify distributed data (i.e. ID3, C4.5, C5,nearest neighbour).Ying-hua, Bing-ru, Dan-yang, and Nan (2011).

3 Secure Multiparty Computation

The aim of SMC Lindell and Pinkas (2009) is to enable parties to carry out distributed computing tasks in a secure manner. It is also concerned with the possibility of deliberately malicious behaviour by some adversarial entity. It is assumed that a protocol execution may come under ”attack” by an external entity, or even by a subset of the participating parties. From their Lindell and Pinkas (2009) study, the next subsections describe SMC according to Security Properties and Adversarial Power.

3.1 Security Properties

Some different definitions for SMC have been proposed. These definitions aim to ensure important security properties that capture most multiparty computation tasks. Lindell and Pinkas (2009) describes these properties:

- Privacy: No party should learn anything more than its prescribed output. The only information that should be learned about other parties’ inputs is what can be derived from the output itself.
- Correctness: Each party is guaranteed that the output that it receives is correct.
- Independence of Inputs: Corrupted parties must choose their inputs independently of the honest parties’ inputs.
- Guaranteed Output Delivery: Corrupted parties should not be able to prevent honest parties from receiving their output. The adversary should not be able to disrupt the computation by carrying out a ”denial of service” attack.

- Fairness: Corrupted parties should receive their outputs if and only if the honest parties also receive their outputs

3.2 Adversarial Power

The adversary can control a subset of participating parties in the protocol. It is necessary to formalise how adversary corruption strategy and what kind of adversary's behaviour can occur.

3.2.1 Corruption Strategy

1. Static corruption model: The adversary has a fixed set of parties under his control. Honest parties remain honest, and corrupted parties remain corrupted.

2. Adaptive corruption model: Rather than having a fixed set of corrupted parties, adaptive adversaries can corrupt parties during the computation. The choice of who, and when to corrupt, can be arbitrarily decided by the adversary. Once a party is corrupted, it remains corrupted from that point on.

3.2.2 Adversary Behavior

1. Semi-honest adversaries: In this model, even corrupted parties correctly follow the protocol specification. However, the adversary obtains the internal state of all the corrupted parties and attempts to use this to learn information that should remain private. This kind of adversaries is also called ”honest-but-curious” and ”passive”.

2. Malicious adversaries: In this adversarial model, the corrupted parties can arbitrarily deviate from the protocol specification, according to the adversary's instructions. Malicious adversaries are also called ”active”.

4 PPDM and Secure Multiparty Computation

A truly secure SMC protocol Vaidya and Clifton (2004) doesn't reveal any information other than its input and output or any information polynomially computable from it, but care in performing these tasks is required, like this, itself might be a privacy breach. This section presents some of the proposed protocols and algorithms in PPDM and SMC proposed by Lindell and Pinkas (2009), Tassa and Gudes (2012), Bogdanov, Niitsoo, Toft, and Willemson (2012), and Teo, Lee, and Han (2012). Notice that, in this section, only three protocols/algorithms are listed.

4.1 Lindell and Pinkas - Lindell and Pinkas (2009)

In their paper, Lindell and Pinkas (2009) present two protocols that can be used as basic building blocks for secure protocols:

- Oblivious Transfer, which security rest on the decisional Diffie-Hellman (DDH) assumption; and
- Oblivious Polynomial Evaluation, based on homomorphic encryption (the authors consider this protocol secure in the semi-honest model and achieves privacy in the face of a malicious adversary).

4.1.1 Oblivious Transfer

This is the protocol described by Lindell and Pinkas (2009):

- **Input:**The sender has a pair of strings (m_0, m_1) and receiver has the bit σ.
- **Auxiliary input:** The parties have the description of a group G of order n, and a generator g for the group; the order of the group is known to both parties.
- **The Protocol**
  1. The receiver R chooses $a, b, c \in_R \{0, ..., n-1\}$ and computes γ and a generator g for the group; the order of the group is known for both parties;
     (a) if $\sigma = 0$ then $\gamma = (g^a, g^b, g^{ab}, g^c)$
     (b) if $\sigma = 1$ then $\gamma = (g^a, g^b, g^c, g^{ab})$
     R send γ to S.
  2. Denote the tuple γ received by S by (x, y, z_0, z_1). Then, S checks that $z_0 \neq z_1$. If they are equal, it aborts outputting $\perp$. Otherwise, S chooses random $u_0, u_1, v_0, v_1 \in_R \{0, ..., n-1\}$ and computes the following four values:

     $w_0 = x^{u_0}.g^{v_0} \qquad k_0 = (z_0)^{u_0}.y^{v_0}$

     $w_1 = x^{u_1}.g^{v_1} \qquad k_1 = (z_1)^{u_1}.y^{v_1}$

     S then encrypts m_0 under k_0 and m_1 under k_1. For the sake of simplicity, assume that one-time pad is used. That is, assume that m_0 and m_1 are mapped to elements of G. Then, S computes $c_0 = m_0.k_0$ and $c_1 = m_1.k_1$ where multiplication is in the group G.

     S sends R the pairs (w_0, c_0) and (w_1, c_1).

3. R computes $k_\sigma = (w_\sigma)^b$ and outputs $m_\sigma = c_\sigma.(k_\sigma)^{-1}$.

4.2 Tassa and Gudes - Tassa and Gudes (2012)

They devised secure distributed protocols for obtaining k-anonymized and l-diverse views of shared databases. They presented two SMC protocols: the first one is a simple SMC protocol for the computing the sum of private integers. Moreover, the second SMC protocol computes the Least Common Ancestor of private nodes in a tree. The LCA proposed protocol is a contribution of the authors.

4.2.1 Sum of Private Integers

Protocol to secure computation of the sum presented by Tassa and Gudes (2012):

- **Input:**Player i, $1 \leq i \leq m$, has an input bit $a_i \in \mathbb{N}$
- **Output:** $a = \sum_{i=1}^{m} a_i$.

```
Player 1 sets $a = 0$.
for $i = 1, ..., m$ do
    Player $i$ generates a random element $r_i \in \mathbb{Z}_N$ and sends to Player $i+1$ (or Player 1 when $i = m$) the value $a = a + a_i + r_i$, where all operations are made in $\mathbb{Z}_N$.
end for
for $i = 1, ..., m$ do
    Player $i$ sends to Player $i+1$ (or Player 1 when $i = m$) the value $a = a - r_i$.
end for
The value of $a$ at this stage is $a = \sum_{i=1}^{m} a_i$.
```

Where N is any sufficiently large integer that is agreed among the players in advance.

4.2.2 Secure Computation of *AND*

This is the protocol to computes securely *AND* proposed by Tassa and Gudes (2012):

- **Input:**Player i, $1 \leq i \leq m$, has an input bit $b_i \in \{0, 1\}$
- **Output:** $b = \prod_{i=1}^{m} b_i$.

```
1. Player 1 sets $a = 0$.
2. for $i = 1,...,m$ do
3.     Player $i$ generates a random element $r_i \in \mathbb{Z}_{m+1}$ and sends to Player $i+1$ (or Player 1 when $i = m$) the value $a = a + b_i + r_i$, where all operations are made in $\mathbb{Z}_{m+1}$.
4. end for
5. for $i = 1,...,m-2$ do
6.     Player $i$ sends to Player $i+1$ the value $a = a - r_i$.
7. end for
8. Player $m-1$ computes $u := a - r_{m-1} = \sum_{i=1}^{m} b_i + r_m$.
9. Player $m$ computes $v = m + r_m$.
10. Player $m-1$ and $m$ output $b = 1$ if $u = v$, and $b = 0$ otherwise.
```

The protocol is based on the fact that the *And* of all bits equals 1 *iff* their sum equals m. Hence, the players execute the secure sum protocol up to one step before its completion (steps 1–8). Then, in steps 9–10, the last two players check whether the sum equals or not. To do that, they need to securely compare two values (u and v), without disclosing them.

4.2.3 Sequential clustering for k-anonymization in horizontally partitioned databases

The distributed sequential clustering protocol in the horizontal partitioning setting proposed by Tassa and Gudes (2012):

- **Input:**m tables $D^i = \{R_1^i, ..., R_{n_i}^i\}$, integer k.
- **Output:** A k-anonymized table, $\overline{D} = \{\overline{R_1}, ..., \overline{R_n}\}$ of $\bigcup_{i=1}^{m} D^i$, where $\sum_{i=1}^{m} n_i$.

```
1. Compute $n = \sum_{i=1}^{m} n_i$.{SMC protocol}
2. Choose a random partition of the data records into $t := \lfloor n/k_0 \rfloor$ clusters, $C_1, ..., C_t$.
3. Player 1 computes the size, closure, and generalization cost of all clusters, $|C_s|, \overline{C}_s$, and $F(\overline{C}_s), i \leq s \leq t$. {SMC protocol}
4. for $i = 1,...,m$ do
```

```
5.      for $j = 1,...,n_i$ do
6.          Let $C_s$ be the cluster to which record $R^i_j$ currently belongs. Compute the
            closure and generalization cost of $C'_s := C_s \backslash \{R^i_j\}$.{SMC protocol}
7.          for $r = 1,...,t, r \neq s$ do
8.              Compute the closure and generalization cost of $C'_r := C_r \bigcup \{R^i_j\}$.
9.              Compute the change in the overall information loss if $R^i_j$ would
                move from $C_s$ to $C_r$:
                $\Delta_{(i,j):s\to r} := (|C'_s|.F(\overline{C'_s}) + |C'_r|.F(\overline{C'_s})) - (|C_s|.F(\overline{C_s}) + |C_r|.F(\overline{C_s}))$.
10.         end for
11.         Let $C_{r_0}$ be the cluster for which $\Delta_{(i,j):s\to r}$ is minimal.
12.         if $|C_s| = 1$ then
13.             Move $R^i_j$ from $C_s$ to $C_{r_0}$ and update the size, closure and general-
                ization cost of $C_{r_0}$.
14.             Remove $C_s$ from the list of clusters.
15.         else
16.             if $\Delta_{(i,j):s\to r} < 0$, move $R^i_j$ from $C_s$ to $C_{r_0}$ and update the size, closure
                and generalization cost of both $C_s$ and $C_{r_0}$.
17.         end if
18.     end for
19.     Transfer to the next player the updated sizes and closures of all clusters.
20. end for
21. for each $C_s$ of size $|C_s| > k_1$ do
22.     Player 1 creates a new cluster and sends a message to all players to move a
        random half of the records in $C_s$ to the new cluster.
23.     Player 1 computes the size, closure, and generalization cost of $C_s$ and the new
        cluster.{SMC protocol}
24. end for
25. if at least one record was moved during the last loop (Steps 4-20), go to Step 4.
```

```
26. **while** the number of clusters of size smaller than $k$ is greater than 1 **do**
27.     Compute the distance between every pair of small clusters,
        $dist(C_s,C_r) := (|C_s \bigcup C_r|.F(\overline{C_s \bigcup C_r})) - (|C_s|.F(\overline{C_s}) + |C_r|.F(\overline{(C_r)})$.
28.     Unify the two closest small clusters.
29. **end while**
30. If there exists a single cluster of size less than $k$, unify it with the cluster to which it is closest.
31. Compute the $k$-anonymization that corresponds to the final clustering. {SMC protocol}
```

The protocol is based on the fact that the *And* of all bits equals 1 *iff* their sum equals m. Hence, the players execute the secure sum protocol up to one step before its completion (steps 1–8). Then, in steps 9–10, the last two players check whether the sum equals or not. To do that, they need to securely compare two values (u and v), without disclosing them.

4.3 Bogdanov, Niitsoo, Toft, and Willemson (2012)

They describe new protocols in the Sharemind model for secure multiplication, share conversion, equality, bit shift, bit extraction, and division.

4.3.1 Resharing Protocol $[\![u]\!] \leftarrow Reshare([\![u]\!])$

- **Data:** Shared value $[\![u]\!]$.
- **Result:** Shared value $[\![w]\!]$ such that $w = u$, all shares w_i are uniformly distributed and u_i and w_j are independent for $i, j = 1,2,3$.

1. $\mathscr{P}_1$ generates random $r_{12} \leftarrow \mathbb{Z}_{2^n}$.
2. $\mathscr{P}_2$ generates random $r_{23} \leftarrow \mathbb{Z}_{2^n}$.
3. $\mathscr{P}_3$ generates random $r_{31} \leftarrow \mathbb{Z}_{2^n}$.
4. All values $*_{ij}$ are sent from $\mathscr{P}_i$ to $\mathscr{P}_j$.
5. $\mathscr{P}_1$ computes $w_1 \leftarrow u_1 + r_{12} - r_{31}$.
6. $\mathscr{P}_2$ computes $w_2 \leftarrow u_2 + r_{23} - r_{12}$.
7. $\mathscr{P}_3$ computes $w_3 \leftarrow u_3 + r_{31} - r_{23}$.
8. Return $[\![w]\!]$.

4.3.2 Multiply Two Shared Values Protocol $[\![w']\!] \leftarrow Mult([\![u]\!],[\![v]\!])$

- **Data:** Shared values $[\![u]\!]$ and $[\![v]\!]$.

- **Result:** Shared value $[\![w']\!]$ such that $w' = uv$.

1. $\mathscr{P}_1$ sends u'_1 and v'_1 to $\mathscr{P}_2$.
2. $\mathscr{P}_2$ sends u'_2 and v'_2 to $\mathscr{P}_3$.
3. $\mathscr{P}_3$ sends u'_3 and v'_3 to $\mathscr{P}_1$.
4. $\mathscr{P}_1$ computes $w_1 \leftarrow u'_1v'_1 + u'_1v'_3 + u'_3v'_1$.
5. $\mathscr{P}_2$ computes $w_2 \leftarrow u'_2v'_2 + u'_2v'_1 + u'_1v'_2$.
6. $\mathscr{P}_3$ computes $w_3 \leftarrow u'_3v'_3 + u'_3v'_2 + u'_2v'_3$.
7. Return $[\![w']\!] \leftarrow Reshare([\![w]\!])$.

They also present bit level protocols.

4.3.3 $\overline{[\![p]\!]} \leftarrow PrefixOR(\overline{[\![p]\!]})$

- **Data:** Bitwise shared vector $\overline{[\![p]\!]}$.

- **Result:** The vector $\overline{[\![p']\!]}$ which has the form 00...011...1, where the initial part 00...01 coincides with the vector originally represented by $\overline{[\![p]\!]}$.

```
$l \leftarrow |\overline{[\![p]\!]}|$.
**if** $l = 1$ **then**
    Return $\overline{[\![p']\!]} \leftarrow \overline{[\![p]\!]}$
**else**
    $\overline{[\![p']\!]}^{(l-1,\ldots\lfloor l/2 \rfloor)} \leftarrow PrefixOR(\overline{[\![p]\!]}^{(l-1,\ldots\lfloor l/2 \rfloor)})$.
    $\overline{[\![p']\!]}^{(\lfloor l/2 \rfloor-1\ldots 0)} \leftarrow PrefixOR(\overline{[\![p]\!]}^{(\lfloor l/2 \rfloor-1\ldots 0)})$.
    **for** $i \leftarrow 0$ **to** $\lfloor l/2 \rfloor - 1$ **do**
        $\overline{[\![p']\!]}^{(i)} \leftarrow \overline{[\![p']\!]}^{(i)} \vee \overline{[\![p']\!]}^{(\lfloor l/2 \rfloor)}$.
    Return $\overline{[\![p']\!]}$.
**end**.
```

4.4 Teo, Lee and Han Teo, Lee, and Han (2012)

The authors present four protocols targeted at semi-honest parties.

4.4.1 Secure Scalar Product Protocol

- **Input:** Alice has input vector $x = [x_1, x_2, ..., x_n]^T$ and Bob has input vector $y = [y_1, y_2, ..., y_n]^T$.
- **Output:** Alice and Bob get an output r^a, r^b, respectively, such that $r^a + r^b = x.y$.

```
Alice generates a private and public key pair $(sk, pk)$
Alice send $pk$ to Bob
for $i = 1$ to $n$ do
    Alice send Bob, $c_i = E_{pk}(x_i)$
end for
Bob computes $w = \prod_{i=i}^{n} c_i^{y_i}$
Bob generates a random plaintext $r^b$
Bob sends to Alice, $w' = w.E_{pk}(-r^b)$
Alice computes $r^a = D_{sk}(w') = x.y - r^b$
```

4.4.2 Secure Matrix Multiplication Protocol

- **Input:** Alice has private d x N matrix A and Bob has private N x n matrix B.
- **Output:** Alice obtains private matrix M^a and Bob obtains private matrix M^b such that their sum $M^a + M^b = AB$ yields the product matrix.

```
for $i = 1$ to $d$ do
    for $j = 1$ to $n$ do
        Alice and Bob securely compute the scalar product of vector $a(i, :)$, and
        vector $b(:, j)$. At the end, Alice and Bob each hold a private value of $M^a$
        and $M^b$ respectively.
    end for
end for
```

4.4.3 Secure Inverse of Matrix Sum Protocol

- **Input:** Alice has private m x m matrix A and Bob has private m x m matrix B.
- **Output:** Alice obtains private matrix M^A and Bob obtains private matrix M^B such that their sum $M^A + M^B = (A+B)^{-1}$ yields the inverse of the sum of their matrices.

1. Bob randomly generates a non singular m x m matrix P
2. Alice and Bob jointly perform Fast Secure Matrix Multiplication to compute AP, at the end of which, Alice and Bob each obtains S^A, S^B respectively such that $S^A + S^B = AP$.
3. Bob computes $S^B + BP$ and sends it to Alice.
4. Alice computes $S^A + S^B + BP$ (ie,$(A+B)P$), and then its inverse $P^{-1}(A+B)^{-1}$.
5. Bob and Alice jointly perform Fast Secure Matrix Multiplication on P and $P^{-1}(A+B)^{-1}$, at the end of which, Alice and Bob each hold private portions M^A and M^B respectively such that
   $M^A + M^B = P(P^{-1}(A+B)^{-1}) = (A+B)^{-1}$

4.4.4 Fast Secure Matrix Multiplication Protocol

- **Input:** Alice has private d x N matrix A and Bob has private N x n matrix B.
- **Output:** Alice obtains private matrix M^a and Bob obtains private matrix M^b such that their sum $M^a + M^b = A + B$ yields the the product matrix.

```
Alice encrypts his/her matrix $E(A)$ and send it to Bob.
for $i = 1$ to $d$ do
    for $j = 1$ to $n$ do
        Bob individual computes $\prod_{k=1}^{N}[E(a(i,k))]^{b(k,j)} \text{x} E(-r^{B}_{i,j})$, where $-r^{B}_{i,j}$ is
        a random number and sends all $E(r^{B}_{i,j})(m\text{x}n)$ back.
        Alice decrypts and obtain $r^A$
        Alice and Bob each hold a private value of $M^a$ and $M^b$.
    end for
end for
```

5 Final Considerations

The work of Lindell and Pinkas (2009) presents a very detailed description and validation of protocols. Despite describing the protocol, they do not present how to use it in PPDM effectively. They also introduced the use of SMC for data mining by constructing a privacy-preserving ID3 classification algorithm

In their paper, Tassa and Gudes (2012) present two SMC protocols for PPDM. They also show how to integrate the SMC (4.2.3)protocol to make a sequential clustering for *k*-anonymisation in horizontally partitioned databases. The resulting approach applies to both horizontal and vertical partitioning scheme, the only cryptographic primitives needed in their approach are an SMC protocol for computing sums and a secure hash function. The presented/proposed protocols are not perfectly secure in the cryptographic sense, as pointed by the authors, they ”leak very little and benign information”.

Bogdanov, Niitsoo, Toft, and Willemson (2012) present several SMC protocols, but as Lindell and Pinkas (2009), do not show a practical way to integrate their protocols with PPDM.

Teo, Lee, and Han (2012), present four SMC protocols, but like Bogdanov, Niitsoo, Toft, and Willemson (2012) and Lindell and Pinkas (2009), no practical way to integrate their protocols with PPDM is showed.

Vaidya and Clifton (2004) states that the advantage of a SMC-based solution is that it gives a better notion of exactly what is revealed. In a perfect SMC protocol, nothing should be revealed, but in the real world, this is not feasible. However, the SMC theory provides ways to delineate what is known and what remains secret. The drawback on using SMC protocols is inefficiency. Generic SMC protocols are impractical when considering large inputs, and this is typical in data mining.

In Teo, Lee, and Han (2012), the authors conclude their work stating that the Secure Multiparty Computation protocols are not very highly efficient yet in a real world. They posit that the malicious model is more likely similar to a real world in PPDM. In a current definition of security, this model strongly guarantees to minimise the loss of information opposed to any strong adversaries. They also believe that to allow some information leakage to build an efficient and secure protocol is acceptable.

As we can observe in related presented papers, Secure Multiparty Computation for Privacy Preserving Data Mining is a growing area. Some gaps must be filled to take the real benefits of this technologies. Several proposals were shown, but just one, Tassa and Gudes (2012), of them presents how to integrate SMC and PPDM more practically. As shown in presented papers, when using SMC in PPDM one must consider the communication overhead (time) introduced by SMC protocols. Despite this issues, the combination of SMC and PPDM could be an effective solution to privacy concerns in today's Data Mining reality.

References

Dan Bogdanov, Margus Niitsoo, Tomas Toft, and Jan Willemson. High-performance secure multi-party computation for data mining applications. *International Journal of Information Security*, 11(6):403–418, 2012. ISSN 1615-5262. doi: 10.1007/s10207-012-0177-2. URL `http://dx.doi.org/10.1007/s10207-012-0177-2`.

Aris Gkoulalas-Divanis and Vassilios S. Verykios. An overview of privacy preserving data mining. *Crossroads*, 15(4):6:23–6:26, June 2009. ISSN 1528-4972. doi: 10.1145/1558897.1558903. URL `http://doi.acm.org/10.1145/1558897.1558903`.

Ninghui Li, Tiancheng Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115, April 2007. doi: 10.1109/ICDE.2007.367856.

Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):59–98, 2009. URL `http://repository.cmu.edu/cgi/viewcontent.cgi?article=1004&context=jpc`.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217302. URL `http://doi.acm.org/10.1145/1217299.1217302`.

Durgesh Kumar Mishra, Purnima Trivedi, and Samiksha Shukla. A glance at secure multiparty computation for privacy preserving data mining. *International Journal on Computer Science and Engineering*, 1(3):171–175, 2009. ISSN 0975–3397.

B Murugeshwari, C Jayakumar, and K Sarukesi. Secure multi party computation technique for classification rule sharing. *International Journal of Computer Applications*, 55(7):1–10, October 2012. doi: 10.5120/8764-2683. Published by Foundation of Computer Science, New York, USA.

Nivetha.P.R and Thamarai selvi.K. A survey on privacy preserving data mining techniques. *International Journal of Computer Science and Mobile Computing*, 2(10):166–170, 2013. ISSN 2320-088X.

Tamir Tassa and Ehud Gudes. Secure distributed computation of anonymized views of shared databases. *ACM Trans. Database Syst.*, 37(2):11:1–11:43, June 2012. ISSN 0362-5915. doi: 10.1145/2188349.2188353. URL `http://doi.acm.org.ez47.periodicos.capes.gov.br/10.1145/2188349.2188353`.

S.G. Teo, V. Lee, and Shuguo Han. A study of efficiency and accuracy of secure multiparty protocol in privacy-preserving data mining. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 85–90, March 2012. doi: 10.1109/WAINA.2012.90.

Jaideep Vaidya and Chris Clifton. Privacy-preserving data mining: Why, how, and when. *IEEE Security and Privacy*, 2(6):19–27, November 2004. ISSN 1540-7993. doi: 10.1109/MSP.2004.108. URL `http://dx.doi.org/10.1109/MSP.2004.108`.

Pingshui Wang. Survey on Privacy Preserving Data Mining. *International Journal of Digital Content Technology and its Applications*, 4(9):1–7, 2010. URL www.aicit.org/jdcta/ppl/01_JDCTAS19-559003.pdf.

Liu Ying-hua, Yang Bing-ru, Cao Dan-yang, and Ma Nan. State-of-the-art in distributed privacy preserving data mining. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 545–549, May 2011. doi: 10.1109/ICCSN.2011.6014329.

Recebido: 06/09/2017
Aprovado: 19/10/2017