An Efficient and Privacy-preserving Similarity Evaluation For Big Data Analytics

Zakaria Gheid, Yacine Challal

► To cite this version:

Zakaria Gheid, Yacine Challal. An Efficient and Privacy-preserving Similarity Evaluation For Big Data Analytics. IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Dec 2015, Limassol, Cyprus. 10.1109/UCC.2015.46 . hal-01308786

HAL Id: hal-01308786

<https://hal.science/hal-01308786>

Submitted on 28 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Efficient and Privacy-preserving Similarity Evaluation For Big Data Analytics

Zakaria Gheid

Ecole nationale supérieure d'informatique
Laboratoire des Méthodes de Conception des Systèmes
Algiers, Algeria
Email: z_gheid@esi.dz

Yacine Challal

Ecole nationale supérieure d'informatique
Laboratoire des Méthodes de Conception des Systèmes
Centre de Recherche sur l'Information Scientifique et Technique
Algiers, Algeria
Email: y_challal@esi.dz

Abstract—Big data systems are gathering more and more information in order to discover new values through data analytics and depth insights. However, mining sensitive personal information breaches privacy and degrades services' reputation. Accordingly, many research works have been proposed to address the privacy issues of data analytics, but almost seem to be not suitable in big data context either in data types they support or in computation time efficiency. In this paper we propose a novel privacy-preserving cosine similarity computation protocol that will support both binary and numerical data types within an efficient computation time, and we prove its adequacy for big data high volume, high variety and high velocity.

Keywords—big data, data analytics, cosine similarity, privacy.

I. INTRODUCTION

Advances in sensing and storing technologies along with the expansion of the number of connected objects have led to capture and store a huge amount of data. This data explosion phenomenon was called big data by Cox et al.[1] and has since been largely shared between academia, industry and the media. According to a Gartner Group report, a Big data involves data sets that are big in size with a high variety of data types and a high velocity of streaming in and out [2]. This widely quoted definition was followed by developing more scalable tools and technologies, such as Google Map Reduce [3] and Apache Hadoop [4], in order to better deal with these new data dimensions. Furthermore, data analysts are struggling upstream to extract patterns and knowledge from datasets combined as never before. Businesses that seek new paradigms, governments that would like enhance their authorities, security agencies and research communities are still rummaging through big data silos to satiate their desires. For example, eBay.com had implemented a 40PB Hadoop cluster to improve its recommendation system. Facebook and Twitter are handling billions of queries which are stored for being analysed with data mining techniques [5]. Another anecdotal example consists of Barack Obama's 2012 winning re-election campaign, in which, big data analytics were used to discover political interest of the voters, thus address them more accurately [6]. However, when individual records involve sensitive information, the application of some data mining techniques such as similarity measurement raises confidentiality issues and breaches privacy policies of services and institutions. For instance, when a hospital receives a new patient, requesting for similar Electronic Medical Record (EMR)[7] to other collaborator hospitals may breach the patient privacy. Concerning the

biometric identity, disclosing a passenger biometric template to be compared with a template list of suspect persons is a privacy breach for innocent passengers. With regard to document similarity, a paper reviewer who searches for similar paper submitted to other journals in order to check for plagiarism shouldn't disclose papers still in submission, otherwise, the journal privacy policy is broken. Moreover, when a service provider is comparing a user sessions to enhance recommendation, mining the sessions in a clear way breaches the user privacy and demean the service reputation.

In this paper we tackle the issue of privacy in big data analytics. In particular, we focus on the cosine similarity metric as a similarity-based method, and we will show that big data requires efficient similarity computation to scale to large amounts of data. Therefore, protecting privacy should not induce high computation overhead. Then we propose a novel cosine similarity calculation method for which we will demonstrate privacy preservation and efficiency.

We organize our paper as follows: Section II presents a background on cosine similarity computation and its involved privacy issue. In Section III we expose recent research works having targeted the privacy concern of scalar product and we show briefly their inadequacy for big data analytics. Then, we introduce a novel efficient and privacy preserving cosine similarity protocol in section IV, and we detail its implementation within two different contexts. Section V shows some real world application scenarios of our protocol, and the last two sections are dedicated to prove the privacy preservation of our protocol as well as its efficiency. We conclude by summarizing the contribution of this paper and its suitability for big data analytics.

II. BACKGROUND AND PROBLEM STATEMENT

Cosine similarity is a statistical metric used for measuring similarities between vectors of numerical attributes. It consists of calculating the cosine ($\cos$) of the angle (θ) between two vectors of attributes. $\cos(\theta)$ belongs to $[-1,1]$ and the more it is closer to 1, the more vectors are similar, and vice-versa. Let $\vec{a} = (a_1, \dots, a_n)$ and $\vec{b} = (b_1, \dots, b_n)$ be 2 vectors in R^n vector space and θ the angle between them. Cosine similarity between $\vec{a}$ and $\vec{b}$ is measured by

$$\cos(\theta) = \frac{(\vec{a} \cdot \vec{b})}{\|\vec{a}\| \|\vec{b}\|} \quad (1)$$

where $(\vec{a} \cdot \vec{b})$ is the dot product obtained as follows

$$(\vec{a} \cdot \vec{b}) = \sum_{i=1}^n (a_i \times b_i) \quad (2)$$

and $\|\vec{a}\|$ (resp. $\|\vec{b}\|$) is the Euclidean norm obtained by $\sqrt{\sum_{i=1}^n a_i^2}$ (resp. $\sqrt{\sum_{i=1}^n b_i^2}$). If vectors are normalized, i.e. $\vec{a} = (a_1/\|\vec{a}\|, \dots, a_n/\|\vec{a}\|)$ and $\vec{b} = (b_1/\|\vec{b}\|, \dots, b_n/\|\vec{b}\|)$, then the cosine similarity will be shortened to the dot product, i.e. $\cos(\theta) = (\vec{a} \cdot \vec{b})$. Figure 1 illustrates the relationship between $\cos(\theta)$ and the similarity of $(\vec{a}, \vec{b})$.

Figure 1. Cosine similarity[8]

Nevertheless, computing the dot product i.e. $\sum_{i=1}^n (a_i \times b_i)$ needs to disclose attributes of one individual to the other. Moreover, in big data era, almost relevant analytics raise from sharing datasets between several parties or from outsourcing them to expert analytics service providers. However, when attributes include sensitive information, disclosing them to other parties threatens clients confidentiality and breaches privacy policies. The issue is then how to compute the similarity of two individuals without revealing their attributes values. Thereafter, we note a cosine similarity computation system by $\chi_1 = \{S, D\}$, where S includes stations involved in the computation and D is the set of required data.

A. Big data analytics requirements

In addition to privacy need, big data analytics require some constraints due to its main characteristics which are illustrated in Figure 2.

1) *Volume requirement*: big data is a great amount of datasets. Therefore, this high volume drives us to compute similarity for many times. Thus, using a method which performs a privacy-preserving similarity computations without inducing a high computation cost is a keystone requirement.

2) *Variety requirement*: big data silos involve a high variety of data types, both structural and non-structural datasets, including textual, numerical and binary attributes. So, we have to address all this variety in similarity computations.

3) *Velocity requirement*: big data are subjected to a high velocity stream of input and output queries which requires in some contexts real time interactions. Thus, introducing privacy preserving mechanisms should not induce unaffordable computation overhead.

Figure 2. Big data characteristics[9]

III. RELATED WORKS

Computing the scalar product in a privacy way has attracted many contributions from the research community. Jaideep Vaidya and Chris Clifton[10] have addressed the problem of association rule mining in vertically partitioned data. In order to identify valid association rules from different transactions distributed over sources, sites need to collaborate provided that they shouldn't disclose their individual transaction data. Thus, authors proposed a privacy preserving scalar product protocol which was based on algebraic operations in order to scale well to data mining problem. Nonetheless, B. Goethals et al.[11] have identified some attacks against this protocol proving that one of the two parties running the protocol[10] can disclose the private input of the other party with probability near to 1. Authors highlighted the privacy risk with $\{0,1\}$ values and affirmed that remedies proposed by[10] have no security proof. In contrast, B. Goethals et al. proposed another secure protocol based on Homomorphic encryption[11]. Nevertheless, Homomorphic encryption induces a high computation overhead according to [12], thus it is inadequate for big data analytics. Wenliang Du and Mikhail J. Atallah[13] introduced a privacy-preserving statistical analysis protocol for cooperative environments. In addition to use Homomorphic encryption operations that raises inefficiency, this protocol is unsecure and can be subjected to privacy breaches when it comes to binary data as proved by[11]. Work proposed by Wei Jiang et al.[14], later extended by Mummoorthy Murugesan et al.[15], has introduced two cosine similarity based-methods for similar document detection. The first method used the matrix-based protocol proposed in[10] which presents privacy risk as aforementioned. The second method has used a Homomorphic encryption-based protocol. Methods introduced by Hiroaki Kikuchi et al.[16] have been proposed for secure remote biometrics authentication. Authors have evaluated similarity using the cosine correlation and the Euclidean distance by implementing two Homomorphic encryption-based protocols. I. Leontiadis et al.[17] have proposed a mechanism for privacy preserving clustering using the cosine similarity metric. The idea of solution consists of applying a rotation in a two dimensional space to generate sub-vectors from the original data vectors. Sub-vectors are then randomly scaled and subjected to cosine computation. Even this technique is efficient, it relies on the average between sub-vectors similarities which is different from direct cosine computation. Dexin Yang et al.[18] have proposed an ElGamal encryption-based protocol for secure cosine similarity computation which resists to malicious

adversaries attacks. Authors have proved that their protocol can target integer attributes unlike proposed schemes which consider, only, similarity of binary data. More recently, R. LU et al. [12] have proposed a privacy preserving cosine similarity protocol for big data analytics. Their solution was based on algebraic operations in order to be more efficient in big data contexts. Nevertheless, this protocol presents a serious privacy risk when it comes to binary data. Indeed, we suppose that Sa and Sb are two stations having respectively $\vec{a} = (a_1, \dots, a_n)$ and $\vec{b} = (b_1, \dots, b_n)$ as individuals. The protocol assumes that Sb sends to Sa a parameter $B = (\sum_{i=1}^n b_i^2)$. So, when it comes to binary attributes: $b_i^2 = b_i \Rightarrow B = (\sum_{i=1}^n b_i)$. Therefore Sa can know the number of $(b_i = 0)$, and the number of $(b_i = 1)$ which is a privacy breach according to Definition 4 presented in section VI-A. Authors have argued that they add the values $(a_n + 1 = a_n + 2 = b_n + 1 = b_n + 2 = 0)$ to ensure the existence of at least two random numbers in the information sent by Sb to Sa which involves bi attributes so that Sa can't guess $\vec{b}$. Though, in the case of binary vectors and when all $(b_i \neq 0)$, Sb will send $(B = n)$ which discloses $\vec{b}$ despite the presence of random numbers in the information sent to Sa . This protocol presents a privacy risk for binary contexts, thus it is not suitable for all big data types.

To summarize, existing solutions can be classified into two categories: a) approaches [11, 13, 14, 15, 16, 18] that include encryption-based methods, such as Homomorphic encryption. In spite of enforcing the privacy, encryption decreases the service efficiency, it involves time expensive operations which is not suitable for the high velocity of big data. b) The second approach [10, 12, 14, 15, 17] comprises algebraic-based methods that apply some arithmetic and/or geometric transformations. Despite the low computation cost, almost all those protocols protect privacy in some contexts but not in others, such as binary data. This is not tolerable in the case of big data known for its high variety of data types.

In this paper, we will propose a new protocol for computing the dot product in order to better address cosine similarity as a privacy preserving and efficient building block for big data analytics.

IV. II-CSP: PRIVACY-PRESERVING AND EFFICIENT COSINE SIMILARITY PROTOCOLS

A. Overview of our solution

Our concern will focus upon computing $\cos(\vec{a}, \vec{b})$ in a privacy way, where $\vec{a}$ and $\vec{b}$ are two individuals, which means computing $(\vec{a} \times \vec{b})$ after a normalization process (see section II.A for normalization detail). While taking into consideration the three big data main characteristics (Volume, Velocity and Variety), our solution consists of obfuscating individuals through adding a noise and performing some arithmetic operations allowing cosine similarity calculation. We adopt an arithmetic-based method to be more efficient in addressing big data high velocity. As regards to big data high volume, it implies a high individual size (a great number of attributes) and a high number of individuals. In order to cope with the high size of individuals our solution uses an attribute-independent noise contrary to existing similar approaches [10, 17] that generate a scalar noise for each individual attribute. With

respect to the high number of individuals, we assess similarities of the maximum available set of individuals in the minimum communication steps. By dealing with normalized individuals we are addressing big data high variety, i.e. we ensure computing similarities of binary individuals without communicating $\|\vec{a}\|$ or $\|\vec{b}\|$ which breach their privacy (see section III), thus we are preserving the privacy of both binary and integer individuals.

Let Sa and Sb be two stations having respectively A (a row matrix) and B (a column matrix) set of individuals. To ensure privacy, our solution allow calculating the product $(A \times B)$ without revealing the individuals themselves. Indeed, our method obfuscates A in a matrix product $(M \times A)$ (where M can be considered as a random noise) and sends the result (MA) to Sb . Sb performs $(MA \times B)$ and sends the result matrix (MAB) to Sa which performs $(M^{-1} \times MAB)$ to get the matrix (AB) that contains in each element $AB[i, j]$: $\cos(\vec{a}_i, \vec{b}_j)$. We set some conditions on the dimension of each matrix $(M, A$ and $B)$ in a way that prevents mutual disclosure risks, thus to preserve privacy.

In what follows, we will present details of our solution as well as a three-party variant protocol which is useful in some application scenarios.

B. II-CSP implementation

In order to comply with different application scenarios, we will propose two protocols. The first is a two-party computation protocol used for a privacy shared cosine similarity computation between two stations. The second protocol takes into consideration analytic service providers, thus, it performs privacy computation between three parties. Let us consider Sa and Sb two stations having respectively $\vec{a}_i = (a_1, \dots, a_n)$ and $\vec{b}_i = (b_1, \dots, b_n)$ individuals. We consider $\chi_1 = \{(Sa, Sb), (M, A, B)\}$ a cosine similarity computation system, where A (resp. B) is a row (resp. column) matrix containing a_i (resp. b_i) individuals and M is an invertible random noise matrix. We define Π , a private and efficient cosine similarity protocol, as follows

$\Pi : [k \times k] \times [k \times n] \times [n \times p] \rightarrow [k \times p]$
 $(M, A, B) \mapsto \Pi(M, A, B) = AB$
Where: $(k, n, p) \in N^* \times N^* \times N^*$ such as:
 $\{1 < k < n, 1 \leq m \leq k, 1 < p < k\}$. The implementation of Π over χ_1 is detailed in Protocol 1.

Protocol 1: Two party II-CSP

Step1: Sa

- 1: Generates a random invertible matrix $M[k \times k]$
- 2: Creates $A[k \times n]$ from m normalized vectors A_i (as rows)
- 3: Adds $(k - m)$ noise rows to A
- 4: Performs $(M \times A)$ and sends the result (MA) to Sb

Step2: Sb

- 5: Creates $B[n \times p]$ from p normalized vectors B_i (as columns)
- 6: Performs $(MA \times B)$ and sends the result (MAB) to Sa

Step3: Sa

- 7: Performs $(M^{-1} \times MAB) = AB[k \times p] = \Pi(M, A, B)$
 - 8: Shares AB with Sb
-

When analytics are performed through an analytic service

provider: P , we can define Π over an augmented cosine similarity computation system: $\chi_2 = \{(Sa, Sb, P), (M, A, B)\}$ in the same way and we detail its implementation over χ_2 in Protocol 2.

Protocol 2: Three party Π -CSP

Step1: P

- 1: Generates a random invertible matrix $M[k \times k]$
- 2: Sends M to Sa

Step2: Sa

- 3: Creates $A[k \times n]$ from m normalized vectors A_i (as rows)
- 4: Adds $(k - m)$ noise rows to A
- 5: Performs $(M \times A)$ and sends the result (MA) to Sb

Step3: Sb

- 6: Creates $B[n \times p]$ from p normalized vectors B_i (as columns)
- 7: Performs $(MA \times B)$ and sends the result (MAB) to P

Step4: P

- 8: Performs $(M^{-1} \times MAB) = AB[k \times p] = \Pi(M, A, B)$
- 8: Shares AB with Sa and Sb

V. EXAMPLES OF REAL-WORLD APPLICATION SCENARIOS

A. Health-care application

To illustrate the application of our protocol in a real-world medical case, we'll introduce this scenario: The clinician-staff of the hospital H_1 receives new patients. After analysis they make some hypothesis about what the patients could have. To check their hypothesis, the clinicians request for local similar cases. Because the result was not relevant enough, they extend their search query for extern cases. The request is handled by a central computation service which multi-cast it for collaborative hospitals. We suppose that the hospital H_1 , the central computation service and a collaborative hospital H_2 correspond respectively to Sb , P and Sa . By applying our three-party Π -CSP, similarities are computed in a privacy way and communicated efficiently to the clinicians. We suppose that an EMR (Electronic Medical Record) has this structure: EMR (High temperature, headache, cough, chest pain, tinnitus, stomach pain). Let P_1 and P_2 be the patients received by H_1 . H_2 's relevant cases are C_1 , C_2 and C_3 which have respectively: hypertension, influenza and stomach problem. We set Parameters: $n = 6$, $k = 2$. Computation details are presented in TableI.

Results have revealed a similar case with a similarity score= 1 between patients $P1(1,1,0,0,1,0)$ and $C1(1,1,0,0,1,0)$, in addition to a relevant one with a similarity score= 0.85 between patients $P2(1,0,1,1,0,0)$ and $C2(1,1,1,1,0,0)$.

B. Recommendation system application

Assume that an online provider of a service S wants to improve his quality of service by setting up a recommendation system. For this, he decides to mine his users' sessions in order to categorize them according to some predefined profile and wants to guarantying their privacy. We suppose that the service includes 7 links: $L_1, \dots, L_7$ and 3 profiles: P_1 , P_3 and P_3 . The service provider will mine the click stream of a user during his last two sessions: S_1 and S_2 to match him with the most

Table I.
HEALTH CARE

	High Temperature	Headache	Cough	Chest pain	Tinnitus	Stomach pain
P1	1	1	0	0	1	0
P2	1	0	1	1	0	0
C1	1	1	0	0	1	0
C2	1	1	1	1	0	0
C3	1	1	0	0	0	1
Normalization						
P1	0.57	0.57	0	0	0.57	0
P2	0.57	0	0.57	0.57	0	0
C1	0.57	0.57	0	0	0.57	0
C2	0.5	0.5	0.5	0.5	0	0
C3	0.57	0.57	0	0	0	0.57
	$M[3 \times 3]$			$M^{-1}[3 \times 3]$		
	2	3	4	2	-0.5	-1
	0	2	4	-3	0	2
	3	5	6	1.5	0.25	-1
	$MA[3 \times 6]$					
	4.92	4.92	1.5	1.5	1.14	2.28
	3.28	3.28	1	1	0	2.28
	7.63	7.63	2.5	2.5	1.71	3.42
	$MAB[3 \times 2]$			$AB[3 \times 2]$		
	6.26	4.51		1	0.32	
	3.74	3.01		0.57	0.85	
	9.67	7.2		0.64	0.32	

suitable profile. Let the service provider and the user be Sa and Sb respectively. Table2 bellow presents the click stream under each link. After applying our two-party Π -CSP in the same way detailed above we obtain results shown in TableII.

Table II.
RECOMANDATION SYSTEM

	L1	L2	L3	L4	L5	L6	L7
P1	1	1	1	1	1	1	1
P2	2	2	0	6	6	0	0
P3	0	0	3	4	5	0	0
S1	0	2	3	6	1	0	7
S2	4	0	2	3	5	3	3
$M[3 \times 3]$				$M^{-1}[3 \times 3]$			
1	2	3		0.33	-0.33	-1	
4	-1	0		1.33	-2.33	-4	
-2	1	1		-0.67	1.67	3	
$AB[3 \times 2]$							
0.7				0.87			
0.5				0.74			
0.53				0.7			

Results show that the most suitable profile for this user is P_1 with asimilarity score= 0.87 for session 1 and 0.74 for session 2. This method can also improve the clustering of the recommendation system by discovering new profiles in the case of divergent similarities.

VI. PRIVACY PROOF

A. security model

In this section we introduce communication and security models that will be used later to proof the privacy preservation property of our proposal.

Definition 1 (Multi-Party Computation MPC): In an MPC model, a given set of participants want to compute the value of a public function relying on the join of their private data. Let $P_1, \dots, P_n$ be the participants and $v_1, \dots, v_n$ their private data respectively. The application of a public function f at the point $(v_1, \dots, v_n)$ i.e. $f(v_1, \dots, v_n)$ is an MPC[19].

Definition 2 (Semi-honest model): In a semi honest model all participants to the execution of a communication protocol are supposed following the protocol with the exception that they save their intermediate results[19].

Definition 3 (Malicious model): In a malicious model there are no suppositions on the behaviour of participants to the execution of a communication protocol. There are three common situations which are: a) participant who stops the protocol untimely, b) participant who substitutes their input and c) participant who refuses to participate[19].

Definition 4 (Privacy preservation): An MPC protocol is privacy-preserving if no participant can learn about his/her collaborators's private data more than he/she can learn from the description of the public function and the result of its application, depending on the considered security model (semi-honest, malicious).

B. System of equations: prerequisites

Definition 5 (System of equations): A system of equations is a collection of equations involving the same set of variables. We set $(k_1, k_2) \in N^* \times N^*$. Let Γ be a system of (k_1) equations including (k_2) variables, then we note Γ as: $\Gamma(k_1, k_2)$.

Lemma 1: the matrix product between $M[k \times k]$ and $A[k \times n]$ is a system of $(k \times n)$ equations that involve $[(k \times k) + (k \times n)]$ variables according to definition 5. i.e.

$$(M \times A) = \Gamma(k \times n, ((k \times k) + (k \times n))) \quad (3)$$

Proposition 1: we set $(k_1, k_2) \in N^* \times N^*$, if $k_1 < k_2$ then $\Gamma(k_1, k_2)$ has infinitely-many solutions.

Proposition 2: we set $(k_1, k_2, k_3, k_4) \in N^* \times N^* \times N^* \times N^*$, if $\Gamma(k_1, k_2)$ and $\Gamma(k_3, k_4)$ are two different systems of equations, then we have

$$\Gamma(k_1, k_2) \cup \Gamma(k_3, k_4) = \Gamma((k_1 + k_3), (k_2 + k_4)) \quad (4)$$

C. Privacy proof of our solution

Theorem 1: Let us consider $\chi = \{(Sa, Sb), (M, A, B)\}$ a cosine similarity computation system and Π our two-party Π -CSP presented in Section IV-B. $\Pi(M, A, B)$ is a privacy-preserving MPC protocol under the malicious model.

Proof: In the construction of the Π -CSP presented in section IV-B we have assumed that:

$$\begin{cases} 1 < k < n \\ 1 < p < k \end{cases}$$

where n, k and p are respectively: the number of individual attributes, the number of Sa 's individuals and the number of Sb 's individuals. We will prove that through these conditions we are preserving the privacy with no additional assumptions on Sa and Sb behaviours while verifying the privacy under the malicious model. Details of privacy proof throughout Π -CSP communication steps are presented below:

Step1: Sb receives $MA[k \times n]$ where M (resp. A) is a $[k \times k]$ (resp. $[k \times n]$) matrix. According to Lemma 1:

$$MA = \Gamma(k \times n, ((k \times k) + (k \times n))) \quad (5)$$

Because we have set $k > 1$, then $(k \times k) > 1 \Rightarrow [(k \times k) + (k \times n)] > [(k \times n) + 1] > (k \times n)$. Therefore, (5) has infinitely-many solutions according to Proposition 1. Thus, Sa 's privacy is preserved.

Step2: Sa receives $MAB[k \times p]$ where B is a $[n \times p]$ matrix. We set:

$$X[k \times n] = MA[k \times n] \quad (6)$$

(6) $\Rightarrow$

$$XB[k \times p] = MAB[k \times p] \quad (7)$$

So, according to Lemma 1

$$XB = \Gamma(k \times p, ((k \times n) + (n \times p))) \quad (8)$$

But as X is well-known by Sa , its $[k \times n]$ variables could be substituted by their values, in this case (8) becomes

$$XB = \Gamma((k \times p), (n \times p)) \quad (9)$$

Then, given (7) and (9) we get

$$MAB = \Gamma((k \times p), (n \times p)) \quad (10)$$

Since we have $k < n$ then $(k \times p) < (n \times p)$ and according to Proposition 1, MAB will have infinitely-many solutions. Therefore Sb 's privacy is preserved.

Step3: $AB[k \times p]$ will be shared to Sb . Because A is a $[k \times n]$ matrix and B is a $[n \times p]$ matrix, we have according to Lemma 1

$$AB = \Gamma(k \times p, ((k \times n) + (k \times p))) \quad (11)$$

But as B is well-known by Sb , its $[k \times p]$ variables could be substituted by their values, thus (11) becomes

$$AB = \Gamma((k \times p), (k \times n)) \quad (12)$$

Since we have $p < n$ then $(k \times p) < (n \times p)$. So, according to Proposition 1, AB will have infinitely-many solutions. Therefore Sa 's privacy is preserved.

Step1 with Step3: Sb will possess (5) from step1 and (11) from step3. Then, according to Proposition 2

$$MA \cup AB = \Gamma(((k \times n) + (k \times p)), ((k \times k) + (k \times n) + (k \times n))) \quad (13)$$

Because variables of $A[k \times n]$ are the same in MA and AB , then the number of variable in (13) will be reduced by $[k \times n]$ variables. Thus, (13) becomes

$$MA \cup AB = \Gamma(((k \times n) + (k \times p)), ((k \times k) + (k \times n))) \quad (14)$$

Because we have set $p < k$ and $(k > 1 \Rightarrow k > 0)$, then we have $(k \times p) < (k \times k) \Rightarrow ((k \times n) + (k \times p)) < ((k \times n) + (k \times k))$ which means according to Proposition 1 that

$MAUAB$ will have infinitely-many solutions. So, Sa 's privacy is preserved. In the cases when noise vectors are added in the matrix A , their product with B will be considered as additional informations derived by Sa about Sb which may breach the Definition 4. Nonetheless, with the conditions that we have set on n , p and k , disclosing B as we have proved is impossible, therefore additional derived information are not helpful.

Under a malicious model we distinguish three possible station behaviours according to Definition 3. if a station stops running the protocol untimely, no privacy risk may occur because the privacy of both stations is well preserved during the three Π -CSP steps as we have proved above. When it comes to injecting fake input by a station, The Π -CSP result will not provide more information about individuals of the other participant station than the cosine similarity result which respects well the Definition 4. Furthermore, if a station refuses to participate to the two-party Π -CSP, then no communication steps will take place.

With the privacy preservation proof throughout the different two-party Π -CSP steps and with no additional assumption on the stations' behaviours, in addition to considering some common malicious behaviours, we have proved the privacy preservation of the two-party Π -CSP under the malicious models according to (Definition 4, Definition 3). ■

Theorem 2: Let us consider $\chi = \{(Sa, Sb, P), (M, A, B)\}$ a cosine similarity computation system and Π our three-party Π -CSP presented in Section IV-B. $\Pi(M, A, B)$ is a privacy-preserving MPC protocol under the semi-honest model.

Proof: We can prove Theorem 2 in the same way we have proved Theorem 1, provided that P follows the protocol and thus it does not disclose M to Sb . By adding this assumption on the behaviour of the service provider P , our three-party Π -CSP is a privacy-preserving MPC protocol under the semi-honest model according to (Definition 4, Definition 2). ■

VII. PERFORMANCE EVALUATION

A. Simulation model and scenarios

In order to prove the efficiency of our Π -CSP we will evaluate the impact of Big data high volume and high velocity on Π -CSP running time and we compare it to the recent privacy-preserving cosine similarity PCSC protocol proposed in [12]. We have avoided to test encryption-based privacy preserving protocols because of their inadequacy for big data analytics as was reported in [12]. We make the experiments on the same set of attributes through a Python script in an Intel i5-2557M CPU running at 1.70GHz and having a 4 GB of RAM. We note that communication costs are not considered.

1) *Volume impact:* Big data high volume implies a high individual size and a high number of individuals. To target these characteristics, we make an evaluation test across three experiments: E_1 , E_2 and E_3 . We plot the running time of the direct cosine similarity computation DCC as a reference experiment in order to make comparison tests more significant. In E_1 we compute the similarity of a set up to 10000 individuals having each one a size of 100 attributes. In E_2 we evaluate the impact of individual size by expanding it to 1000 attributes and computing the similarity of the same set.

Finally, we perform E_3 by re-expanding the size of individuals to 10000 attributes, in order to check the precedent results.

2) *Velocity impact:* In this evaluation we will model the Π -CSP and PCSC protocol using the queuing theory. we consider an analytics service provider that receives on the average λ requests for individual similarity evaluation, every minute, according to a Poisson process. we assume that the request processing times are exponentially distributed with rate μ requests per minute and the service provider is operating all day long. We distinguish two different queuing system models whether their capacity is infinite or finite. Using Kendall notation[20] these models are denoted by $M/M/1$ and $M/M/1/k$ and having both an infinite request sources and a FIFO service discipline. To determine λ and μ parameters, we explore the results of volume evaluation described above and we consider that one request corresponds to two individuals in addition to considering one minute as the time unit. Details are shown in TableIII.

Table III.

QUEUEING MODEL

Requests / min		100	500	1000	5000
number of individuals		200	1000	2000	10000
λ		100	200	500	10000
Size=100	Π -CSP Rtime (s)	0.00020	0.00048	0.00062	0.00260
	PCSC Rtime (s)	1.52	7.31	14.88	74.44
	$\mu - \Pi$ -CSP ($\times 10^7$)	3	6.25	9.67	11.53
	$\mu - \text{PCSC}$	3947	4103	4032	4030
Size=1000	Π -CSP Rtime (s)	0.00051	0.00144	0.00230	0.00975
	PCSC Rtime (s)	14.88	73.70	148.41	749.94
	$\mu - \Pi$ -CSP ($\times 10^7$)	1.18	2.32	2.6	3.08
	$\mu - \text{PCSC}$	403	407	404	400
Size=10000	Π -CSP Rtime (s)	0.00451	0.01343	0.02497	0.07886
	PCSC Rtime (s)	148.41	749.94	1480.30	7480.09
	$\mu - \Pi$ -CSP ($\times 10^6$)	1.33	2.23	2.4	3.8
	$\mu - \text{PCSC}$	40	40	40	40

B. Results and discussion

In what follows we present the different evaluation results and derive conclusions.

1) *Volume evaluation results:* In E_1 we tested the impact of the number of individuals on the similarity computation time. Results presented in Figure 3a show the efficiency of our Π -CSP compared to PCSC. By performing E_2 we have obtained results illustrated in Figure 3b which reveal the efficiency of Π -CSP with a low distance increase ($\simeq 0.0007$ s measured with a set of 10000 individuals) from DCSC running time. However, expanding the size of individuals has expanded significantly the distance between PCSC and DCSC running time with a rate of ($\simeq 348$ s measured with a set of 10000 individuals). The final experiment E_3 illustrated in Figure 3c is confirming the precedent results, Π -CSP running time remains stable as it was in E_2 in the neighborhood of $\exp(-5)$ s (with a set of 10000 individuals) and keeps a slow increasing distance rate from DCSC running time, while PCSC running time has augmented significantly with a rate of ($\simeq 7955$ s measured with a set of 10000 individuals) which increases consequently the distance from DCSC running time to ($\simeq 8103$ s). Volume evaluation tests have confirmed the efficiency of our Π -CSP compared to the recent PCSC protocol, it does not provoke an overhead in

Figure 3. Volume evaluation

running time either by performing a similarity computation on a large set of individuals or after increasing their size. Thus, Π -CSP is more suitable for Big data high volume.

2) *Velocity evaluation results:* Performance of the queueing systems running Π -CSP and PCSC are measured across the server utilization rate U , the mean queue length Q and the mean response time of the system R in function of the traffic intensity factor ρ . Performance results are described according to each model.

The $M/M/1$ model corresponds to a service provider who has an unlimited access. Performance measures of this model are calculated by the following equations

$$\rho = \frac{\lambda}{\mu} \quad (15)$$

$$U = \rho \quad (16)$$

$$Q = N - \rho \quad (17)$$

$$R = \frac{N}{\lambda} \quad (18)$$

Where N is the average number of requests in the system obtained as

$$N = \frac{\rho}{1 - \rho} \quad (19)$$

Results of Table IV illustrated in Figure 4 show that the server running Π -CSP is undergoing a low intensity traffic ρ that raises in a low probability of server overload. This result is confirmed by looking the low server utilization rate U and the very low response time R even for a high request rate. The zero queue length result whatever the traffic intensity is also confirming the efficiency of the server running Π -CSP. However, running PCSC have resulted in a non linear increased intensity of the request traffic ρ that finished by a rate $\rho = 1.24 > 1$ which prevents the system reach the steady state and overload the server with an utilization rate $U = 124\%$, this unreal rate reveal the need to add more PCSC server.

Results of Table V illustrated in Figure 4 are obtained by increasing the size of individuals to 1000 attributes, they reveal the efficiency of Π -CSP server which provides a high speed service by looking the low response time R and the zero queue length Q . However, PCSC server was subjected to an earlier overload reaching a utilization rate $U = 1250\%$ which requires adding more and more server.

After increasing the size of individuals to 10000 attributes, results shown in Table VI and illustrated in Figure 4 show that Π -CSP server is still providing an efficient service, while PCSC server becomes totally overloaded, thus it can not even handle a low traffic rate $\rho = 100$ requests/minute.

Table IV.

M/M/1 EVALUATION RESULTS (SIZE OF INDIVIDUALS= 100)

Requests / min	100	500	1000	5000
ρ Π -CSP ($\times 10^{-4}$)	0.03	0.08	0.1	0.4
ρ PCSC	0.025	0.12	0.25	1.24
U Π -CSP ($\times 10^{-4}$)%	3	8	10	40
U PCSC %	2.5	12	25	124
N Π -CSP	0.00	0.00	0.00	0.00
N PCSC	0.03	0.14	0.33	∞
Q Π -CSP	0.000	0.000	0.000	0.000
Q PCSC	0.001	0.017	0.082	∞
R Π -CSP (μs)	0.033	0.016	0.010	0.008
R PCSC (μs)	259.94	277.55	329.81	∞

Table V.

M/M/1 EVALUATION RESULTS (SIZE OF INDIVIDUALS= 1000)

Requests / min	100	500	1000	5000
ρ Π -CSP ($\times 10^{-3}$)	0.008	0.021	0.038	0.16
ρ PCSC	0.25	1.23	2.47	12.5
U Π -CSP ($\times 10^{-3}$)%	0.8	2.1	3.8	16.2
U PCSC %	25	123	247	1250
N Π -CSP	0.00	0.00	0.00	0.00
N PCSC	0.33	∞	∞	∞
Q Π -CSP	0.00	0.00	0.00	0.00
Q PCSC	0.08	∞	∞	∞
R Π -CSP (μs)	0.08	0.04	0.038	0.032
R PCSC (μs)	3300	∞	∞	∞

Because unlimited request access has produced an overload in the server running PCSP, we have used $M/M/1/k$ system to model less efficient servers who have an access limited to k requests at one time. We have fixed the system capacity k to 300 requests involving individuals of 1000 attributes, and we measured its performance. Also, we introduce the request reject rate λ_r in order to evaluate the quality of service. Results

Figure 4. M/M/1 performance

Figure 5. M/M/1/k performance

Table VI.

M/M/1 EVALUATION RESULTS (SIZE OF INDIVIDUALS= 10000)

Requests / min	100	500	1000	5000
ρ II-CSP ($\times 10^{-2}$)	0.0075	0.02	0.04	0.13
ρ PCSC	2.5	12.5	25	125
U II-CSP ($\times 10^{-2}$)%	0.75	2	4	13
U PCSC %	250	1250	2500	12500
N II-CSP	0.00	0.00	0.00	0.00
N PCSC	∞	∞	∞	∞
Q II-CSP	0.00	0.00	0.00	0.00
Q PCSC	∞	∞	∞	∞
R II-SCP (μs)	0.75	0.45	0.41	0.26
R PCSC (μs)	∞	∞	∞	∞

Table VII.

M/M/1/K EVALUATION RESULTS (SIZE OF INDIVIDUALS= 1000)

Requests / min	100	500	1000	5000
ρ II-CSP ($\times 10^{-3}$)	0.0085	0.021	0.038	0.162
ρ PCSC	0.25	1.23	2.47	12.5
U II-CSP ($\times 10^{-3}$)%	0.85	2.1	3.8	16.2
U PCSC %	25	99.99	99.99	99.99
λ_e II-CSP	100	500	1000	5000
λ_e -PCSC	100	407	405	400
λ_r II-CSP	0	0	0	0
λ_r -PCSC	0	93	595	4600
N II-CSP	0.00	0.00	0.00	0.00
N -PCSC	0.33	296	299	300
Q II-CSP	0.00	0.00	0.00	0.00
Q -PCSC	0.08	295	298	299
R II-CSP (μs)	0.08	0.04	0.038	0.032
R -PCSC (μs)	3333	727304	739319	749782

are obtained as follows

$$\rho = \frac{\lambda}{\mu} \quad (20)$$

$$U = \rho(1 - P_k) \quad (21)$$

$$N = \sum_{n=1}^k k\rho^n P_0 \quad (22)$$

$$Q = N - (1 - P_0) \quad (23)$$

$$R = \frac{N}{\lambda_e} \quad (24)$$

$$\lambda_r = \lambda - \lambda_e \quad (25)$$

Where λ_e , P_0 and P_k are respectively the effective arrival rate, the probability that system is full and the probability that system is empty. they are obtained as follows

$$\lambda_e = \lambda \times (1 - P_k) \quad (26)$$

$$P_i = \rho^i \frac{1 - \rho}{1 - \rho^{k+1}} \quad (27)$$

Results of $M/M/1/k$ model evaluation presented in Table VII and illustrated in Figure 5 show an efficiency in the server running Π -CSP with the same performance as it was in the unlimited access model (with an individual size=1000 attributes). Nevertheless, limiting request access to 300/minute in order to avoid the overload of PCSC server did not improve its performance which is revealed by a high response time R and a maximum utilization rate $U = 99.99\%$ for a high intensity traffic. Running the PCSC system with a full request queue $Q = 299$ has led to a high request reject rate $\lambda_r = 4600$ requests from 5000, which decreases consequently the quality of service.

VIII. CONCLUSION

In this paper we have proposed a novel efficient and privacy-preserving protocol for cosine similarity computation that we have called Π -CSP. Our protocol ensures data privacy in different scenarios through a matrix product. By a privacy proof throughout Π -CSP steps we are protecting the privacy of both numerical and binary attributes in order to target big data high variety. Across different performance evaluation tests, Π -CSP running time remains efficient either for a high data volume or a high velocity stream. When comparing these results to existing constructions, our protocol is the most suitable for big data analytics.

REFERENCES

- [1] Michael Cox and David Ellsworth. Application-controlled demand paging for out-of-core visualization. In *Proceedings of the 8th Conference on Visualization '97, VIS '97*, pages 235–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.
- [2] Doug Laney. Meta group 3d data management: controlling data volume, velocity and variety, 2001.
- [3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [4] The apache hadoop project. url [https : //hadoop.apache.org/](https://hadoop.apache.org/).
- [5] Vignesh Prajapati. *Big data analytics with R and Hadoop*. Packt Publishing Ltd, 2013.
- [6] Daniel Kreiss. *Taking our country back: The crafting of networked politics from Howard Dean to Barack Obama*. Oxford University Press, 2012.
- [7] V. Ilie, J.F. Courtney, and C. Van Slyke. Paper versus electronic: Challenges associated with physicians - usage of electronic medical records. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 142–142, Jan 2007.
- [8] Programering. mahout similarity algorithm. url [http : //www.programering.com/a/mtnxizmwatk.html](http://www.programering.com/a/mtnxizmwatk.html). retrieved 07/09/2015.
- [9] Quora. what is big data and hadoop. url [http : //www.quora.com/what-is-big-data-and-hadoop](http://www.quora.com/what-is-big-data-and-hadoop). retrieved 07/09/2015.
- [10] Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02*, pages 639–644, New York, NY, USA, 2002. ACM.
- [11] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikainen. On private scalar product computation for privacy-preserving data mining. In Choon-sik Park and Seongtaek Chee, editors, *Information Security and Cryptology - ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 104–120. Springer Berlin Heidelberg, 2005.
- [12] Rongxing Lu, Hui Zhu, Ximeng Liu, Joseph K. Liu, and Jun Shao. Toward efficient and privacy-preserving computing in big data era. *IEEE Network*, 28(4):46–50, 2014.
- [13] Wenliang Du and Mikhail J. Atallah. Privacy-preserving cooperative statistical analysis. In *Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual*, pages 102–110, Dec 2001.
- [14] Wei Jiang, M. Murugesan, C. Clifton, and Luo Si. Similar document detection with limited information disclosure. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 735–743, April 2008.
- [15] Mummoorthy Murugesan, Wei Jiang, Chris Clifton, Luo Si, and Jaideep Vaidya. Efficient privacy-preserving similar document detection. *The VLDB Journal*, 19(4):457–475, 2010.
- [16] Hiroaki Kikuchi, Kei Nagai, Wakaha Ogata, and Masakatsu Nishigaki. Privacy-preserving similarity evaluation and application to remote biometrics authentication. In Vicenç Torra and Yasuo Narukawa, editors, *Modeling Decisions for Artificial Intelligence*, volume 5285 of *Lecture Notes in Computer Science*, pages 3–14. Springer Berlin Heidelberg, 2008.
- [17] Iraklis Leontiadis, Melek Önen, Refik Molva, M.J Chorley, and G.B Colombo. Privacy preserving similarity detection for data analysis. In *CSAR 2013, Collective Social Awareness and Relevance Workshop, co-located with the 3rd international conference on Social Computing and its Applications, 30 September-2 October 2013, Karlsruhe, Germany, Karlsruhe, GERMANY, 09 2013*.
- [18] Dexin Yang, Baolin Xu, Bo Yang, and Jianping Wang. Secure cosine similarity computation with malicious adversaries. In Nabendu Chaki, Natarajan Meghanathan, and Dhinakaran Nagamalai, editors, *Computer Networks and Communications (NetCom)*, volume 131 of *Lecture Notes in Electrical Engineering*, pages 529–536. Springer New York, 2013.
- [19] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.
- [20] Erol Gelenbe, Guy Pujolle, and JCC Nelson. *Introduction to queueing networks*, volume 2. Wiley Chichester, 1998.