Private Collaborative Business Benchmarking in the Cloud

Somayeh Sobati-Moghadam
s.sobati@hsu.ac.ir

Amjad Fayoumi
a.fayoumi@lancaster.ac.uk

***Abstract*—Collaborative business benchmarking helps organizations and businesses to evaluate current and future-state goals. Cloud business benchmarking provides valuable insight for organizations to understand and compare their efficiency and effectiveness against peers. However, privacy is the most concerns. We present a privacy-preserving prototype for collaborative benchmarking in the cloud. Sensitive business's data is encrypted and can only be decrypted by a set of parties through a threshold group decryption scheme.**

I. Introduction

Nowadays, the value of merger & acquisition (M&A) has increased significantly. For instance, in 2015, the value of global M&A was 6,012 US billion dollar. Organizations involved in M&A are usually operating within different economic jurisdictions and they follow different standards and performance measures. In such as scenario, information sharing increases due to the need for streamlining standards and performance measures between the companies and their peers, holding umbrella company or subsidiary company. This became more possible and much easier by considering emergent technology such as the cloud computing, which appeals businesses and organizations due to a wide variety of benefits. Cloud computing provides an opportunity for organizations and businesses to share the expertise of their business via database services and also cutting the cost of complex information infrastructure and data management [1]. Beside the advantages such cost saving and service benefits, cloud computing provides higher availability, scalability and more effective disaster recovery rather than in-house operations [2].

Benchmarking is an important process for companies and businesses to stay competitive in the marketing [3], especially during financial crisis businesses have to take their bearing on the market [4]. Using targeted improvement measures, businesses can evaluate their achievements and performance against their competitors. *Key Performance Indicator* (KPI) is a statistical quantity measuring that evaluates the success and the performance of an organization or of a particular activity in which it engages [5]. Small businesses might want to cooperatively forecast the future of their business to make more informed decisions [6]. Outsourcing business information to the cloud and collaborative benchmarking would empower smaller businesses to decision making via a valuable broader information shared by largest businesses and organizations. As a result, organizations and enterprises have great interest to share their PKIs via the cloud to enhance collaborative benchmarking and extract interesting knowledge [7]. Outsourcing such data has some advantages in term of cost reducing and using the benefits of cloud computing while extracting useful business information using some *evaluation functions* (e.g. the functions like SUM, AVERAGE, ...) over shared PKIs.

However, since business's data is a very valuable asset for many organizations and business, data confidentiality and privacy is the most concern. Because organizations or individuals do not want to reveal their private data for various legal and competitive reasons. Highly confidential data about a company's operations are sensitive and privacy should be preserved. A trivial solution is to encrypt data before sending them to the cloud.

Consider a scenario in which a group of companies U that are interested in comparing their KPIs without revealing sensitive data to others. Each company we call as a party, U_i, has a set of KPIs, $x_{i1}, ..., x_{in}$ that are encrypted and sent to a *cloud service provider (SP)*. Each party shares his encrypted inputs and other parties including the SP will learn nothing except what can be inferred from their inputs and the final output [3]. In addition to eliminating storage and computation at the party's, the goal is to compute a selected function over encrypted shared data at the SP's. Once a function is commonly chosen, the SP computes the function over encrypted data. The final result must be decrypted by a set of parties, each of whom shared his private data [8] through executing a privacy-preserving protocol. Data privacy should be preserved in such a way that other parties and the SP learn nothing about the shared data.

Data privacy can be preserved using Multi Party Computation (MPC). In MPC, computations are executed by the parties, which induces computational overhead at the user's and is not compatible with cloud outsourcing scenarios. Several privacy-preserving benchmarking have been proposed [9]–[11] but, cannot fit privacy requirements in the cloud context. In this paper, we introduce a prototype for collaborative evaluating and benchmarking of businesses over encrypted data shared in the cloud. Then, the encrypted result is sent back to a trusted server called *proxy server*. A group of parties executes a *threshold group decryption* scheme to decrypt the result. The proxy server plays a role of dealer for key distribution and a combiner in the threshold group decryption. We assume that it does not maliciously collaborate with the SP [12]. We assume that all parties are interested in obtaining the correct results, thus they submit the correct data. Another important aspect of this prototype is that the parties only communicate with the SP, but never amongst them. Anonymity among the parties can only be achieved, if they do not need to address messages to others [12].

The rest of this paper is structured as follows. In Section II, we describe some preliminaries. In Section III we present our privacy preserving benchmarking. We analyze the security of our solution in Section IV and we conclude the paper in Section V.

II. Preliminaries

A. Homomorphic Encryption

Fully Homomorphic Encryption (FHE) is an encryption scheme enables performing arbitrary computations over encrypted data without decryption. The first FHE scheme introduced by Gentry [13], which is intensive and inefficient to implement. Partial Homomorphic Encryption (PHE) is more efficient and closer to practical solution. PHE allows computing addition or multiplication over encrypted data, but not both.

The Paillier cryptosystem [14] is the most efficient PHE in the literature. The Paillier cryptosystem provides homomorphic properties as follows [12], [15]:
For two plaintexts M_1 and M_2,

$$E(M_1+M_2) = E(M_1)\times E(M_2) \quad andE(k\times M) = E(M)^K$$

such properties enable computing over encrypted data without decryption.

B. Threshold Group Decryption

A threshold group decryption scheme allows any subset of $t+1$ out of l parties to decrypt a ciphertext, but disallows the decryption if less than t parties participate [16]. Threshold group decryption may work when some parties are corrupted and do not play according to their nominal behavior.

For a group of users U, a threshold group decryption scheme $TGD_{\langle U\rangle}$, consists of four algorithms (`KG`, `En`, `parDec`, `Com`) described below.

- **Key generation** `KG`: takes as input the number of parties ℓ, the threshold parameter t, and a random string w. It outputs a public key PK and a list of private keys $sk_1, ..., sk_\ell$.
- **Encryption algorithm** `En`: takes as input the public key PK, a random string w and a cleartext M. It outputs a ciphertext c.
- **Partial decryption algorithm** `parDec`: takes as input the public key PK, an index i where $1 \leq i \leq \ell$, the private key sk_i and a ciphertext c. It outputs a decryption share c_i.
- **Combining algorithm** `Com`: takes as input a public key PK, a ciphertext c, a list of partial decryption $c_1, .., c_{t+1}$ and outputs a cleartext M.

We can consider that up to t party could be corrupted by a passive or active attack. In passive, an adversary eavesdrops the parties. By contrast, an active adversary controls the behavior of corrupted party. A threshold decryption is $t-robust$ if the combiner (proxy) be able to decrypt any ciphertext even if t parties corrupt [16].

C. Threshold Group Decryption of Paillier Cryptosystem

The threshold group decryption of Paillier cryptosystem is defined as below [16]:
If ℓ in the number of parties then $\Delta = \ell!$

- `KG`:
  Choose an integer n, product of two strong primes p and q, such that $p = 2p' + 1$ and $q = 2q' + 1$ and $gcd(n, \varphi(n)) = 1$, then set $m = p'q'$. Let β be a random element in $\mathbb{Z}_n^*$. Then the proxy randomly chooses $(a, b) \in \mathbb{Z}_n^* \times \mathbb{Z}_n^*$ and sets $g = (1 + n)^a \times b^n \bmod n^2$. The secret key $SK = \beta m$ is shared with Shamir's secret sharing scheme: let $a_0 = \beta m$, t random values a_i in $\{0, ..., n \times m - 1\}$ are chosen and set $f(x) = \sum_{i=0}^{t} a_i X^i$. The secret key sk_i of i^{th} party U_i is $f(i) \bmod nm$. The public key PK consists of g, n and the value $\theta = L(g^{m\beta}) = am\beta \bmod n$.
- `En`:
  To encrypt a message M first a random element, x, is chosen $x \in \mathbb{Z}_n^*$ and compute $c = g^M x^n \bmod n^2$.
- `parDec`:
  The i^{th} party U_i computes the decryption $c_i = c^{2\Delta sk_i} \bmod n^2$ using his secret key sk_i.
- `Com`:
  If less than t parties shares their partial decryption, this step fails. Let S be a set of $t + 1$ partial decryption shared, the plaintext computes as:

  $$M = L\left(\Pi_{j\in S} c_j^{2\mu_{0,j}^S} mod \quad n^2\right) \times \frac{1}{4\Delta^2\theta} \quad mod \quad n$$

  where

  $$\mu_{0,j}^S = \Delta \times \prod_{j'\in S\setminus\{j\}} \frac{j'}{j'-j} \in \mathbb{Z}$$

III. Privacy Preserving Business Benchmarking (PPBB)

Let consider the collaborative business benchmarking and forecasting in which a set of parties denote by $U = \langle U_1, U_2, ..., U_\ell\rangle$, want to store their data in a cloud and delegate the administration of data, to a specialized SP. Ones they agree to enhance an evaluation function, they execute $PPBB$ to send selected function and getting the result. In our setting, a trusted proxy server plays the role of the dealer and the combiner, which combines the partial results and broadcasts the results to all parties.

$PPBB$ executes in three different phases: Initialization phase (A) for generating a public and a set of secret keys and sending to the parties. Phase (B) for encrypting and storing data at the SP side, and phase (C) for sending the selected function and get the results. The parties are unaware of the other parties or the number of all parties. The SP evaluates an arbitrary function over encrypted data. In decryption phase, minimum t number of parties and the proxy interact and cooperate to decrypt the results. The parties need not be online at all during the bulk of the computation, they are online only when it's the time to decrypt the results [17]. The parties do not interact between them, they send their partial decryption to the proxy.

A *Initialization*: The proxy server executes the `KG` algorithm and generates a public key PK and ℓ secret keys $sk_1, sk_2, \ldots, sk_\ell$ and sends (PK, sk_i) to party U_i , $i = 1, \ldots, \ell$.

B *Storing Data*: In this phase, which is off-line [18], a party sends and stores data at the SP side. Data is encrypted using the public key PK and sent to the SP. To simplicity, we write this phase as: $c \leftarrow En(PK, M)$.

C *Querying Data*: In this phase, which is online, a subset of parties $U_T = \langle U_1, U_2, ..., U_{t+1} \rangle$ first choose a query.

- step1 : The proxy, encrypts the query using the PK and sends the encrypted query to the SP.
- step2 : The SP performs the query on the ciphertexts $encRes \leftarrow Qur(C)$, and sends the results $encRes$ to the proxy. The proxy broadcasts to the parties the $encRes$
- step3 : A set of parties $U_T = \langle U_1, U_2, ..., U_{t+1} \rangle$ run `parDec`, via their individual secret keys, compute $c_i \leftarrow parDec(encRes, sk_i)$, and send their partial results c_i to the proxy.
- step4 : The proxy combines the partial decryption to compute the message $M \leftarrow Com(c_1, c_2, ..., c_t, c_{t+1})$, then it broadcasts M to all parties.

IV. Security Analysis

In PPBB, sensitive data is encrypted by a public key and the SP never receives secret keys $sk_i \quad i = 1, \ldots, \ell$, so the SP never sees sensitive data, ensuring data privacy. Similarly, only a set of t parties along with the proxy server can run a partial decryption algorithm to decrypt the result, which guarantees data privacy even when t parties collude.

The Paillier cryptosystem provides strong security guarantees, called *semantic security* [13]. Semantic security implies that any adversary seeing ciphertexts cannot learn any information about the underlying plaintext. In the other word, ciphertexts stored at the SP side leak no information about the sensitive data.

V. Conclusion

This paper describes a privacy-preserving protocol for collaborative benchmarking in the cloud. Data privacy is guaranteed by computing over encrypted data. The result of computation can be decrypted only when a group of parties jointly compute a threshold group decryption. This work can be extended in a number of ways. Future directions include:

- The proposed solution provides data privacy against passive adversaries. It is necessary to improve security against active adversaries by adding some authentication mechanisms such as Message authentication codes (MAC) or digital signature. Such cryptographic schemes are used to detect any changes that an active adversary makes.
- We would like to implement the proposed solution and compare with the state-of-the-art solutions in terms of computational overhead and security guarantees.

References

[1] E. Shi, J. Bethencourt, T.-H. Chan, D. Song, and A. Perrig, "Multi-dimensional range query over encrypted data," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 350–364.

[2] E. Damiani, S. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Key management for multi-user encrypted databases," in *Proceedings of the 2005 ACM workshop on Storage security and survivability*. ACM, 2005, pp. 74–83.

[3] F. Kerschbaum, "Secure and sustainable benchmarking in clouds," *Business & Information Systems Engineering*, vol. 3, no. 3, pp. 135–143, 2011.

[4] D. Herrmann, F. Scheuer, P. Feustel, T. Nowey, and H. Federrath, *A Privacy-Preserving Platform for User-Centric Quantitative Benchmarking*. Springer, 2009.

[5] F. Kerschbaum, D. Dahlmeier, A. Schröpfer, and D. Biswas, "On the practical importance of communication complexity for secure multi-party computation protocols," in *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC), Honolulu, Hawaii, USA, March 9-12*, 2009, pp. 2008–2015. [Online]. Available: http://doi.acm.org/10.1145/1529282.1529730

[6] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara, "Private collaborative forecasting and benchmarking," in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. ACM, 2004, pp. 103–114.

[7] L. Xiong, S. Chitti, and L. Liu, "Preserving data privacy in outsourcing data aggregation services," *ACM Transactions on Internet Technology (TOIT)*, vol. 7, no. 3, p. 17, 2007.

[8] Y. Shoham and M. Tennenholtz, "Non-cooperative computation: Boolean functions with correctness and exclusivity," *Theoretical Computer Science*, vol. 343, no. 1, pp. 97–113, 2005.

[9] M. J. Atallah, M. Bykova, J. Li, K. B. Frikken, and M. Topkara, "Private collaborative forecasting and benchmarking," in *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, WPES 2004, Washington, DC, USA, October 28*, 2004, pp. 103–114. [Online]. Available: http://doi.acm.org/10.1145/1029179.1029204

[10] P. Bogetoft, I. Damgård, T. P. Jakobsen, K. Nielsen, J. Pagter, and T. Toft, "A practical implementation of secure auctions based on multiparty integer computation," in *10th International Conference in Financial Cryptography and Data Security, FC 2006, Anguilla, British West Indies, February 27-March*, 2006, pp. 142–147. [Online]. Available: https://doi.org/10.1007/11889663_10

[11] G. Di Crescenzo, *Private Selective Payment Protocols*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 72–89. [Online]. Available: https://doi.org/10.1007/3-540-45472-1_6

[12] F. Kerschbaum, "Practical privacy-preserving benchmarking," in *Proceedings of The Ifip Tc 11 23rd International Information Security Conference*. Springer, 2008, pp. 17–31.
[13] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.
[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptologyEUROCRYPT99*. Springer, 1999, pp. 223–238.
[15] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard, "Practical multi-candidate election system," in *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*. ACM, 2001, pp. 274–283.
[16] P.-A. Fouque, G. Poupard, and J. Stern, "Sharing decryption in the context of voting or lotteries," in *Financial Cryptography*. Springer, 2001, pp. 90–104.
[17] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 1219–1234.
[18] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty computation with low communication, computation and interaction via threshold fhe," in *Advances in Cryptology–EUROCRYPT 2012*. Springer, 2012, pp. 483–501.