International Journal of Advanced Culture Technology Vol.10 No.2 240-245 (2022)
DOI https://doi.org/10.17703/IJACT.2022.10.2.240

IJACT 22-6-32

Similarity measurement based on Min-Hash for Preserving Privacy

[1]Hyun-Jong Cha, [2]Ho-Kyung Yang, [3]You-Jin Song

[1]Dr., Dept. of Multimedia Science, Chungwoon Univ., Korea
[2]Prof., Dvision. of Information Technology Education, Sunmoon Univ., Korea
[3]Prof., Dept. of Information Management, Dongguk Univ., Korea
chj826@kw.ac.kr, porori0421@naver.com, song@dongguk.ac.kr

Abstract

Because of the importance of the information, encryption algorithms are heavily used. Raw data is encrypted and secure, but problems arise when the key for decryption is exposed. In particular, large-scale Internet sites such as Facebook and Amazon suffer serious damage when user data is exposed. Recently, research into a new fourth-generation encryption technology that can protect user-related data without the use of a key required for encryption is attracting attention. Also, data clustering technology using encryption is attracting attention. In this paper, we try to reduce key exposure by using homomorphic encryption. In addition, we want to maintain privacy through similarity measurement. Additionally, holistic similarity measurements are time-consuming and expensive as the data size and scope increases. Therefore, Min-Hash has been studied to efficiently estimate the similarity between two signatures Methods of measuring similarity that have been studied in the past are time-consuming and expensive as the size and area of data increases. However, Min-Hash allowed us to efficiently infer the similarity between the two sets. Min-Hash is widely used for anti-plagiarism, graph and image analysis, and genetic analysis. Therefore, this paper reports privacy using homomorphic encryption and presents a model for efficient similarity measurement using Min-Hash.

Keywords: *Similarity Measurement, MinHash, Homomorphic Encryption, Private MinHash, Privacy Preserving*

1. INTRODUCTION

As areas of representation of data become diverse and storageable, an era is approaching in which numerous objects that exist in the real world can be expressed as data. As a result, applying similarities to objects to traditional methods can only afford the amount of data and computational time. Min-Hash, proposed by Broder, is a kind of locality Sensitive Hashing (LSH) technique that allows the form of a set to be connotated, such as a signature, and gives an approximate estimate of the similarity of the sets. Min-Hash represents the smallest value when two sets of elements are recorded in a particular hash function and is a method that can be used to approximate similarities [1-4].

While encryption has been essential to data security in recent years, traditional encryption technology does not fully protect user-related data due to the exposure of keys due to frequent use of keys. Thus, the homomorphic encryption proposed by Rivest, Addleman and Dertouzous was first proposed a method to perform several operations without a decode key, even when the plain text is encrypted. In addition, Gentry designed a fullly homomorphic encryption based on the challenges of number theory and lattice theory,

Manuscript received: May 25, 2022 / revised: June 2, 2022 / accepted: June 7, 2022
Corresponding Author: song@dongguk.ac.kr
Tel: + +82-54-770-2347, Fax: +82-54-770-2001
Professor, Dept. of Information Management, Dongguk Univ., Korea

enabling computers to perform all computations of addition, subtraction, multiplication, and division of ciphertext. Therefore, in this paper, efficient similarity measurement through Min-Hash can be applied to homomorphic encryption, which is one of the four generations of encryption technology, to maintain privacy. The structure of this study is as follows. Chapter 2 introduces background knowledge and related research, and Chapter 3 introduces Private Min-Hash to which the same type encryption technology as conventional Min-Hash is applied. Chapter 4 describes experiments and results to demonstrate the effectiveness of the method proposed in this paper. Chapter 5 describes the problems and conclusions that need to be resolved in the future.

2. RELATED WORKS

2.1 Similar measurements

Jaccard Similarity is a method of measuring the similarity of two objects represented in a set by calculating the relative magnitude of the intersection with respect to the size of the union of both sets. It can show the degree of similarity [5].

Russell-Lao similarity is similar to jacquard similarity. The method of measuring two object similarity differs from jacquard similarity in that it is the size of the complete set rather than the size of the union of the two sets. Calculate the relative magnitude of the intersections relative to represent the similarity of both sets [6].

The method often used to measure similarity between Jacquard-like and Russell-Lao-like is similar to Jacquard, but methods using Russell-Lao-like similarity are also being studied [7]. There are various methods for measuring the similarity between the two sets, and the similarity can be measured in various ways depending on the situation. In this paper, I would like to calculate the similarity between two objects based on the similarity between Russell and Rao [8].

2.2 Min-Hash

Min-Hash is a method that shows the smallest result value when two sets of elements are put into a specific hash function in history, and can be used to approximate the similarity. The smallest result value that comes out via Min-Hash is expressed as Min-Hash Value, and the Min-Hash Value of the set A is shown in hmin (A). The probability that the two sets of comparison targets A and B have the same Min-Hash Value is the same as the two sets of jacquard similarity [9-11]

The hash function h that is typically used to obtain the Min-Hash Value of a set of m elements is $ax + b$ mod p. a and b are arbitrary natural numbers, and p is the smallest prime number greater than or equal to m. Figure 1 shows the concept of Min-Hash.

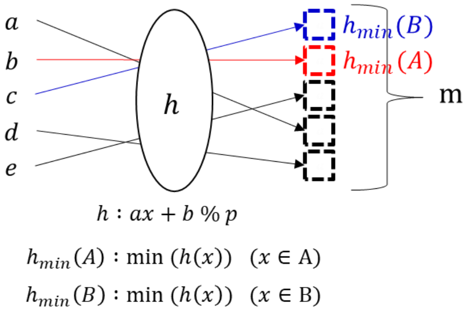

a
b
c
d
e
h
$h_{min}(B)$
$h_{min}(A)$
m
$h : ax + b \% p$
$h_{min}(A)$: min $(h(x))$ $(x \in$ A)
$h_{min}(B)$: min $(h(x))$ $(x \in$ B)

Figure 1. Min-Hash

When there are n Min-Hash Values for a set A via n Min-Hash, it can be expressed in the form of a vector and expressed by Min-Hash Signature. Therefore, n hash functions can be used to calculate the similarity of the generated Min-Hash Signatures of A and B to approximate the similarity of sets A and B.

2.3 Homomorphic Encryption

Homomorphic, which means homomorphism in homomorphic encryption, comes from homomorphism, which is often used in mathematics, and refers to an event (map) that maintains operations between two units of the same type. In other words, homomorphic encryption is a password system that saves specific operations such as addition and multiplication with the idea of associating the elements of plaintext space with the elements of cryptographic space. Among the homomorphic encryptions, the password that can be executed by only some operations is called partial homomorphic encryption. Elgamal passwords on a finite body can only be multiplied. Multiplicative homomorphic encryption. There is additive homomorphic encryption that protects only the addition [12-15].

3. PROPOSED PRIVATE MINHASH

3.1 Basic Structure of the System

The basic structure of this paper is as shown in Figure 2, based on the Russell & Rao similarity comparison system to which the multiplicative homomorphic cipher is applied.

- **Anonymizer.** It performs the calculation requested by the user, but cannot understand the user's information.
- **Requestor.** When you want to know the similarity of datasets, ask the anonymizer.
- **Supporters.** Provides information to the Annoymizer about the information requested by the Requestor.

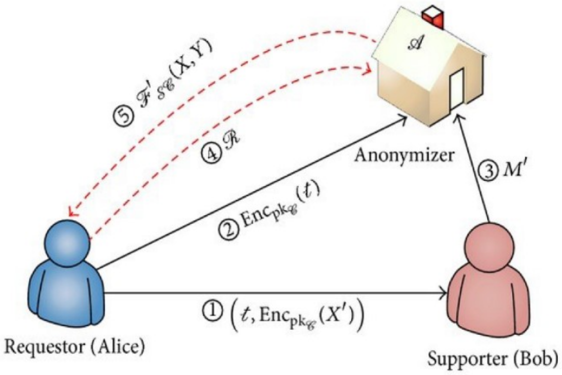

⑤ $\mathscr{F}'_{S\mathscr{C}}(X,Y)$
$\mathscr{A}$
④ $\mathscr{R}$
Anonymizer
③ M'
② $\mathrm{Enc}_{\mathrm{pk}_{\mathscr{C}}}(t)$
① $(t, \mathrm{Enc}_{\mathrm{pk}_{\mathscr{C}}}(X'))$
Requestor (Alice)
Supporter (Bob)

Figure 2. Homomorphic Encryption Scheme

Basically, Annonymizer performs operations on homomorphic encrypted data, and Alice (Requestor) requests a query. The person who provides the data corresponding to the query is made up of Bob (Supporter). For example, assuming Alice has data {2,3} and Bob has {1,2}, Alice has any decimal value to convert her data to a Binary Set. Determine t. Then, from the complete data, only the data that you have that is unlikely to change to 10,000 tons is processed by t-1. After that, the t value and data are encrypted with the public key provided by Anonymizer. Once the encryption is complete, Anonymizer will be provided with the encrypted t-value, and Bob will provide the t-value and Alice's encrypted data. Bob is similar in method to Alice, but when converting his data to a Binary Set, he processes the data he has with t2 and the other data with t. Finally, after representing the encrypted data M via multiplication with the encrypted data received from Alice, the requested query without providing information about Alice and Bob's data to Anonymizer. Provides M` with randomly shuffled M data so that can be executed.

Figure 3 shows the requester and supporter dataset.

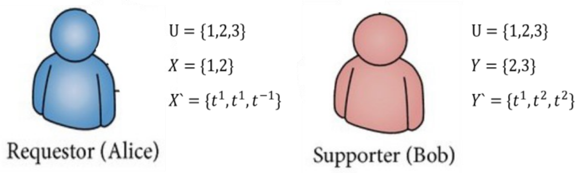
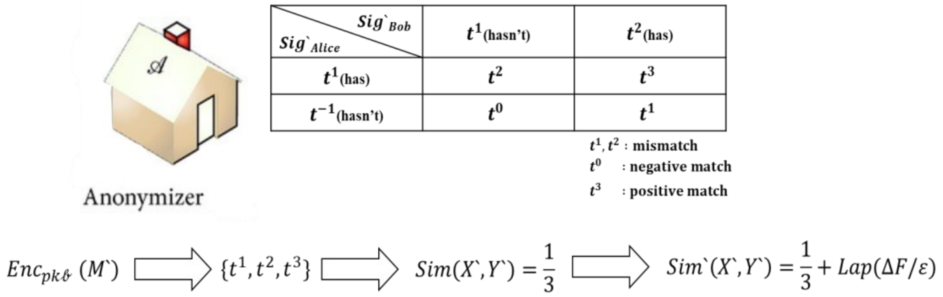


Figure 3. Requester and supporter Dataset


Anonymizer
$Enc_{pk\mathcal{A}}\ (M`) \Rightarrow \{t^1, t^2, t^3\} \Rightarrow Sim(X`, Y`) = \frac{1}{3} \Rightarrow Sim`(X`, Y`) = \frac{1}{3} + Lap(\Delta F/\varepsilon)$

$Sig`_{Bob}$ / $Sig`_{Alice}$	t^1(hasn't)	t^2(has)
t^1(has)	t^2	t^3
t^{-1}(hasn't)	t^0	t^1

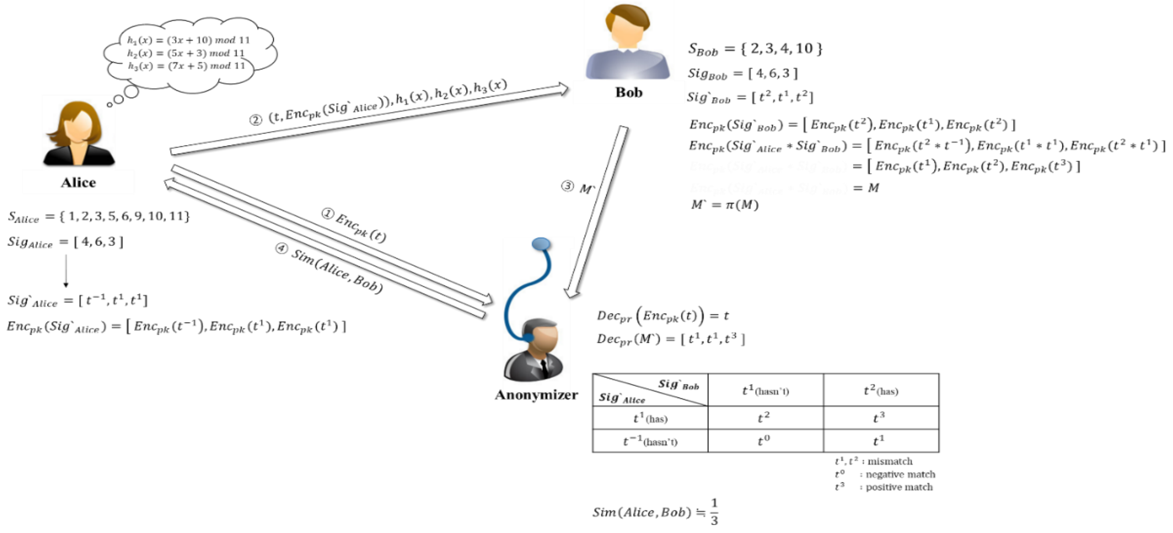
t^1, t^2 : mismatch
t^0 : negative match
t^3 : positive match

Figure 4. Anonymizer

After receiving the encrypted t-value and M` from Alice and Bob, Anonymizer calculates the similarity as shown in Figure 4, which attempts to decrypt its own private key and converts it into plaintext data. Then, the data to be transmitted to Alice generates an arbitrary noise value according to the confidentiality (Sensitivity) of the data, inserts it into the calculated similarity value, and transmits it. By generating and adding random noise values, no one can know exactly about the actual similarity between Alice and Bob's data while the similarity is calculated.

In this paper, a procedure to efficiently calculate similarity using Min-Hash is applied, and the calculated similarity value is an approximation. So, we can pass the calculated value to Alice immediately without adding any noise values.

3.2 Similarity Measurement Model


$h_1(x) = (3x + 10)\ mod\ 11$
$h_2(x) = (5x + 3)\ mod\ 11$
$h_3(x) = (7x + 5)\ mod\ 11$
Alice
② $(t, Enc_{pk}(Sig`_{Alice})), h_1(x), h_2(x), h_3(x)$
Bob
③ $M`$
① $Enc_{pk}(t)$
④ $Sim(Alice, Bob)$
Anonymizer
$S_{Alice} = \{1,2,3,5,6,9,10,11\}$
$Sig_{Alice} = [4,6,3]$
$Sig`_{Alice} = [t^{-1}, t^1, t^1]$
$Enc_{pk}(Sig`_{Alice}) = [Enc_{pk}(t^{-1}), Enc_{pk}(t^1), Enc_{pk}(t^1)]$
$S_{Bob} = \{2,3,4,10\}$
$Sig_{Bob} = [4,6,3]$
$Sig`_{Bob} = [t^2, t^1, t^2]$
$Enc_{pk}(Sig`_{Bob}) = [Enc_{pk}(t^2), Enc_{pk}(t^1), Enc_{pk}(t^2)]$
$Enc_{pk}(Sig`_{Alice} * Sig`_{Bob}) = [Enc_{pk}(t^2 * t^{-1}), Enc_{pk}(t^1 * t^1), Enc_{pk}(t^2 * t^1)]$
$= [Enc_{pk}(t^1), Enc_{pk}(t^2), Enc_{pk}(t^3)]$
$= M$
$M` = \pi(M)$
$Dec_{pr}\left(Enc_{pk}(t)\right) = t$
$Dec_{pr}(M`) = [t^1, t^1, t^3]$
$Sim(Alice, Bob) \fallingdotseq \frac{1}{3}$

$Sig`_{Bob}$ / $Sig`_{Alice}$	t^1(hasn't)	t^2(has)
t^1(has)	t^2	t^3
t^{-1}(hasn't)	t^0	t^1

t^1, t^2 : mismatch
t^0 : negative match
t^3 : positive match

Figure 5. Similarity Measurement Model

In this paper, we utilize Min-Hash utilizing the data in the set to apply efficient similarity measurement of a large set. The overall configuration procedure consists of four stages as shown in Figure 5 [16].

- **Step 1.** Alice sets an arbitrary minority t value. Then, an arbitrary hash function is defined to obtain a complete set of Min-Hash Values. If Alice has the specified hash function Min-Hash Value, it converts it to t, otherwise it converts it to t^{-1}. Alice who generated k hash functions will have a signature $Enc_{pk}(Sig_{Alice})$ indicating k Min-Hash Values. Anonymizer is encrypted with the provided public key and tells Bob the t-value and signature $Enc_{pk}(t, Sig_{Alice})$ like a hash function. Anonymizer provides an encrypted t-value.

- **Step 2.** Bob creates a signature via the hash function provided in ② of Figure 5, and then creates and encrypts the signature in the same way as Alice. It then multiplies the values of the two encrypted signatures to generate a new signature M. When passing to Anonymizer, it provides data M' that randomly shuffles the data belonging to M to execute the query when it does not provide the information of M data.

- **Step 3.** In ③ of Figure 5, the Anonymizer obtains $Dec_{pr}(M')$ obtained by decrypting M' provided by Bob with its own private key. Anonymizer stores the data corresponding to t^3 among the data in the score set.

$$Dec_{pr}(M') = [\, t^1, t^1, t^3 \,]$$
$$Score = \{ x \mid x \in Dec_{pr}(M`) \ and\ x = t^3 \}$$

The Boolean Table shown in Table 1 determines the type of similarity for the four variables. The element that both objects have in common is a positive match. In this paper, this variable is represented by t3. Elements that have only one of both objects are represented by t1 and t2 in mismatch. Finally, elements that both objects do not have are represented by t0 in a negative match.

Table 1. Boolean Table

Sig_{Bob} Sig_{Alice}	t^1(hasn't)	t^2(has)
t^1(has)	t^2	t^3
t^{-1}(hasn't)	t^0	t^1

t^1, t^2 : mismatch
t^0 : negative match
t^3 : positive match

Alice and Bob similarity Sim (Alice, Bob) calculated through the signature transmitted by Anonymizer is calculated via the number of four variables Alice and Bob similarity is calculated as follows.

$$Sim(Alice, Bob) = \frac{|Score|}{|Dec_{pr}(\mathrm{M}`)|}$$

- **Step 4.** Alice is provided with an approximation of the similarity calculated in ④ in Figure 5 to Anonymizer.

4. CONCLUSION

In this paper, we introduced a customer segmentation model to measure the data similarity of smart devices that protect personal information using homomorphic encryption and Min-Hash. Experiments have shown that the clustering introduced in this paper is not as good as clustering with older encryption in terms of clustering quality, but more efficient in terms of speed. Therefore, it can be applied to a real-time similar analysis model that guarantees customer privacy

REFERENCES

[1] Lee, JeeYoung, “A study on research trend analysis and topic class prediction of digital transformation using text mining,” *International journal of advanced smart convergence,* Vol. 8, No. 2, pp. 183-190, 2019.

[2] Jung, Soo-Mok, “Image Watermarking Algorithm using Spatial Encryption,” *The Journal of the Convergence on Culture Technology,* Vol. 6, No. 1, pp. 485-488, 2020.

[3] Hahm, Sangwoo, and Linlin Chen, “The Role of Professors' Intellectual Stimulation for Intellectual Growth among Chinese Students Who Study in Korea: The Moderating Effect of Growth Need Strength,” *International Journal of Advanced Culture Technology*, Vol. 8, No. 3, pp. 45-53, 2020.

[4] Seifoddini, Hamid, “A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications,” *The international journal of production research*, Vol. 27, No. 7, pp. 1161-1165, 1989.

[5] Welke, Pascal, Tamás Horváth, and Stefan Wrobel, “Min-hashing for probabilistic frequent subtree feature spaces,” in *International Conference on Discovery Science*, Springer, Cham, pp.67-82, Oct, 2016.

[6] Russell, Paul F., and T. Ramachandra Rao, “On habitat and association of species of anopheline larvae in south-eastern Madras,” *Journal of the Malaria Institute of India*, Vol. 3, No. 1, 1940.

[7] Sneath, P. H., and R. R. Sokal, “Numerical taxonomy,” *Bergey’s manual of systematic bacteriology 1*, pp. 39-42, 2006.

[8] Choi, Seung-Seok, Sung-Hyuk Cha, and Charles C. Tappert, “A survey of binary similarity and distance measures,” *Journal of systemics, cybernetics and informatics*, Vol. 8, No. 1, pp. 43-48, 2010.

[9] Chum, Ondrej, James Philbin, and Andrew Zisserman, “Near duplicate image detection: Min-hash and TF-IDF weighting,” in *Bmvc*. Vol. 810. pp. 812-815, Sep, 2008.

[10] Lee, David C., Qifa Ke, and Michael Isard, “Partition min-hash for partial duplicate image discovery,” in *European Conference on Computer Vision,* Springer, Berlin, Heidelberg, pp. 648-662, Sep, 2010.

[11] Koslicki, David, and Hooman Zabeti, “Improving minhash via the containment index with applications to metagenomic analysis,” *Applied Mathematics and Computation*, Vol. 354, pp. 206-215, 2019

[12] Tsiounis, Yiannis, and Moti Yung, “On the security of ElGamal based encryption,” in *International Workshop on Public Key Cryptography*. Springer, Berlin, Heidelberg, pp. 117-134, Feb, 1998.

[13] Pan, Miao, Jinyuan Sun, and Yuguang Fang, “Purging the back-room dealing: Secure spectrum auction leveraging paillier cryptosystem,” *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 4, pp. 866-876, 2011.

[14] Jeong, Yunsong, Joon Sik Kim, and Dong Hoon Lee, “Privacy-Preserving k-means Clustering of Encrypted Data,” *Journal of the Korea Institute of Information Security & Cryptology*, Vol. 28, No. 6, pp. 1401-1414, 2018.

[15] Almutairi, Nawal, Frans Coenen, and Keith Dures, “K-means clustering using homomorphic encryption and an updatable distance matrix: secure third party data clustering with limited data owner interaction,” in *International Conference on Big Data Analytics and Knowledge Discovery,* Springer, Cham, pp. 274-285, Aug, 2017

[16] Syropoulos, Apostolos, “Mathematics of multisets,” in *Workshop on Membrane Computing,* Springer, Berlin, Heidelberg, pp. 347-358, Aug, 2000.