International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Secure Multiparty Equality Check Based on Homomorphic Cryptosystem

Rashid Sheikh, Durgesh Kumar Mishra

Abstract: ***There exist many scenarios where multiple parties jointly work on some common project but these parties are not interested to show actual value of data to each other. Here we propose an algorithm where multiple cooperating but distrustful parties can compare their data for equality without disclosing it to one another. This is an improvement over two party comparison algorithm devised earlier by other researchers. The proposed work is suitable for semi honest adversaries who respect rules of the protocol but somehow try to know private values with other parties.***

Keywords: ***Secure multiparty computation, homomorphic cryptosystem, semi honest adversaries.***

I. INTRODUCTION

Sensitive data may be distributed among multiple sites during data mining operation with a restriction to maintaining privacy. The disclosure of these sensitive data is not allowed but still the parties needs to cooperate during computation. This problem where many parties work on some common task but they are concern about privacy of their sensitive data is called Secure Multiparty Computation (SMC) [3, 4]. Many privacy-preserving methods have been suggested [5] which will be described in the next section. Three types of adversarial nature of the parties are considered namely honest adversary, semi honest adversary, and malicious adversary. An honest adversary follows the steps of the protocol and never attempt to know the sensitive data of the participating parties. A semi honest also known as honest-but-curious adversary follows the steps of the protocol but may attempt to receive sensitive data of participating parties. A malicious adversary neither follows the rules of the protocol nor honours the privacy of others. The protocols in the honest adversary model are simplest and cheapest while with the malicious adversary model the protocols are most complex, inefficient and expensive.

There exists two SMC architectures namely real SMC model and ideal SMC model. There exists a Trusted Third Party (TTP) in the ideal model of SMC which accepts private data from the participating parties, evaluates common function and distributes the result to the cooperating parties. All the parties are supposed to have full trust in the TTP. Ideal Model is easier to implement but it is expensive due to the cost involved for maintaining the TTP.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Rashid Sheikh, Research Scholar at Computer Science and Engineering Department, Mewar University, Chittorgarh, India, Associate Professor at Acropolis Institute of Technology and Research Indore, India,
Durgesh Kumar Mishra, Computer Science and Engineering Department, Sri Aurobindo Institute of Technology, Indore, India,

Since the private data is provided to the TTP, the privacy is preserved. In actual practice some government agency or some agency approved by the government may work as TTP. In case of any dispute the TTP may help resolve it in the court of law. But along with the behaviour of the adversaries the behaviour of the TTP is also significant. If the TTP becomes malicious whole scheme fails to provide accuracy and privacy. This will violate both the goals of SMC. There is no TTP in the real model of SMC. The participating parties follow certain steps as per the protocol to compute common function their private data. This model is slightly complex to implement as the data is to be shared directly with other parties in such a way that its actual value must not be visible to these parties. Another problem with real model is that in case of dispute the absence of the TTP makes the decision making more difficult. But it is cost effective model. In our proposed protocol we use additive homomorphic encryption which can be defined as follows:

Let PU be the public key and PR be the private key. An additive homomorphic encryption scheme must fulfil following conditions:

If d_1 and d_2 be the data then

$E_{PU}(d_1+d_2) = E_{PU}(d1) + E_{PU}(d_2)$

For a constant k :

$E_{PU}(k.d) = k.E_{PU}(d)$

In our work we used additive homomorphic cryptosystem as proposed by Pallier in 1999 [2]. The whole scheme can be described as below:

Key Generation: choose two prime numbers p and q such that $p < q$ and p does not divide q-1.

Public key PU= p.q=n

Private key PR = {n, λ}

Where λ is the LCM of p-1 and q-1.

Encryption with PU:

$E_{PU}(d) = (1+n)d \cdot r^n \mod n^2 = c$

Where r is a random number

Decryption with PR:

$d= \frac{(c^{\lambda} \bmod n^2)-1}{n}.\lambda^{-1} \bmod n$

Proving that the Pallier's scheme is additive homomorphic:

$= E_{PU}(d_1) \times E_{PU}(d_2) \bmod n^2$

$= \left((1+n)^{d_1}.r_1^{\,n}\,(1+n)^{d_2}.r_2^{\,n}\right) \bmod n^2$

$=(1+n)^{d_1+d_2}.(r_1r_2)^n \bmod n^2$

$= E_{PU}(d_1+d_2)$

II. LITERATURE SURVEY

Protocols are available in the literature for two party equality comparisons [1] based on Pallier's homomorphic cryptosystem. Two Parties P0 and P1 with the data d0 and d1 as shown in Fig.1 want to know whether d0 and d1 are equal without revealing their private data to one another.

The two-party comparison algorithm we call it EqualityCheck algorithms which be explained with the following steps:

EqualityCheck algorithm

Assume two parties P_0 and P_1 having their private data d_0 and d_1 respectively want to know whether $d_0 = d_1$.

Step1: The party P_0 generates public key pair (PU and PR) using Paillier's homomorphic cryptosystem where PU is the public key and PR is private key.

Step2: The party P_0 computes encryption of its data using PU and sends (PU, $E_{PU}(d_0)$) to the party P_1.

Step3: The party P_1 selects a random number r and computes $[E_{PU}(d_0) – E_{PU}(d_1)].r$, and sends to P_0.

Step4: The party P_0 decrypts the expression $[E_{PU}(d_0) – E_{PU}(d_1)].r$ with the private key PR as below

$D_{PR}\ [E_{PU}(d_0) – E_{PU}(d_1)].r$

$D_{PR}[r.E_{PU}(d_0 - d_1)]$ using homomorphic property

$r.(d_0 - d_1)$

If the result of the decryption is 0, then both the data are equal else not.

Step4: The party P_0 shares the result with P_1.

Privacy of both the data is preserved. The party P_1 does not know d_0 as it cannot decrypt due to lack of private key PR. Similarly, the party P_0 cannot learn d_1 as it doesn't know random number r. But, the can easily know whether their data are equal or not.

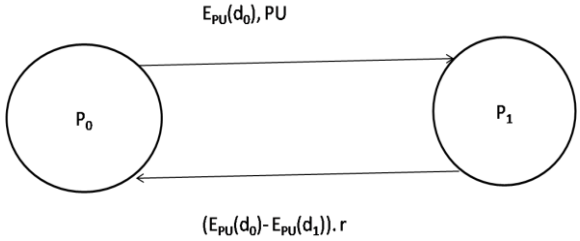

Fig. 1. Two-party comparison algorithm

The above algorithm is suitable for honest but curious or semi honest parties who follow the rules of the protocol but may try to infer secret data of the other party. Flowchart for *Equality Check* algorithm is shown in the Fig.2.

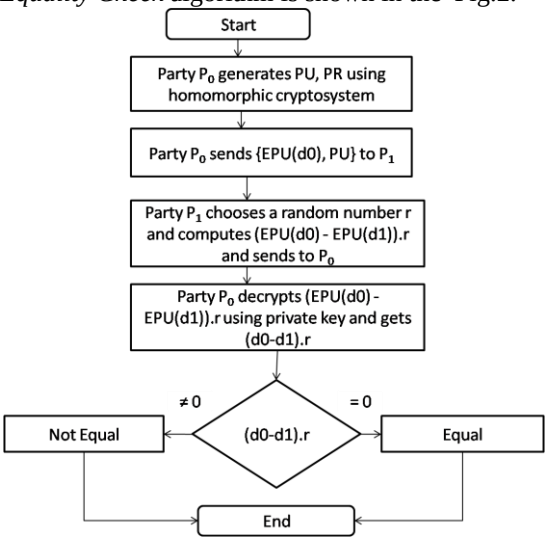

Fig. 2. Flowchart for Equality Check algorithm

III. PROPOSED PROTOCOL FOR MULTIPARTY COMPARISON

In this section we propose a protocol for comparison of data of multiple parties for equality without revealing data of one party to other parties.

A. Informal Description

The multiple parties having their secret data will be arranged in a ring. One of the parties will initiate the protocol. It will generate public-private key pair using Paillier's homomorphic cryptosystem using public key cryptography. It will run two-party algorithm with the next party in the ring as described by the Algorithm 1 . If the equality holds the protocol may proceed for the next party in the ring. If equality doesn't hold at any point in the ring, the result may be declared as Equality Doesn't Hold in multiparty. If the protocol reaches at the last party in the ring and equality still holds, the result may be declared as Equality Holds in multiparty case.

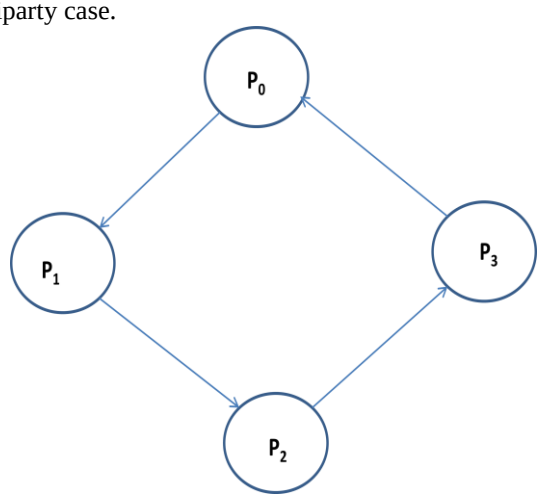

Fig. 3. Secure Multiparty Comparison

B. Formal Description

Consider parties P_0 to P_{k-1} are arranged in a ring as depicted in Fig.3.These parties contain their secret data d_0 to d_{k-1}. The algorithm of equality comparison for secret data is proposed as *Multi Equality Check* below.

Multi Equality Check

Input: k parties P_0 through P_{k-1} with data d_0 through d_{k-1}.

Requires: All the parties to know whether all data equal or not without disclosing data to one another.

Step1: n = k

Step2: For i = 0 to k-1 do

Party P_i and P_{i+1} Compare d_i and d_{i+1} using *Equality Check*

If $(d_i = d_{i+1})$ then continue

Else write "Not equal"

break

Write "Equal"

End of for loop

Flowchart for *Multi Equality Check* algorithm is shown in the Fig.4.

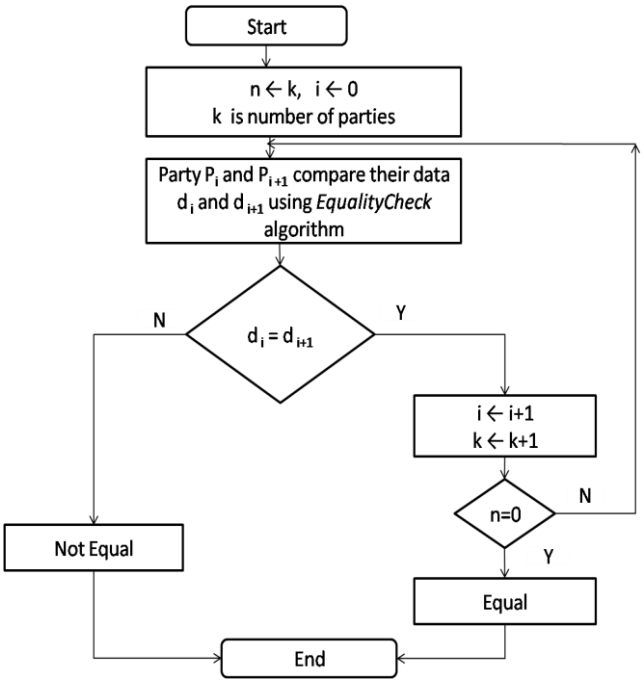

Fig.4. Flowchart for Multi Equality Check algorithm

IV. RESULTS

The privacy preservation objective while comparing the data for equality by multiple joint but distrustful parties is achieved by a physical and logical ring structure. The algorithm *MultiEqualityCheck* is applicable to more than two parties as compared to two party comparisons by *EqualityCheck*. The proposed algorithm and architecture has many benefits over the previous algorithm as listed in the Table-I. Our architecture uses a point-to-point ring network in which parties start comparison of their data pair-by-pair in a single direction. In each of the pairs the respondent party may choose a different random number and the initiator may generate a different public-private key pair.

Table- I: Improvements in Multi Equality Check

S. No.	Criteria	EqualityCheck	MultiEqualityCheck
1.	Number of parties	Two	More than two
2.	Network type	Dedicated between two parties	Point-to-point Ring Network
3.	Random number	Single random number used by respondent	Each pair may use different random number
4.	Key Pair	Single key pair used by initiator party	Each pair of party may use different key pair

The protocol is suitable for semi-ideal adversaries. If the parties follow the rules if the protocol, they will be able to know whether their data are equal or not. In case of equality holding true the privacy will be violated because of obvious reasons.

V. CONCLUSION AND FUTURE SCOPE

We have extended the two-party comparison protocol to multiparty scenario. Our algorithm uses homomorphic cryptosystem using public key cryptography. The originator party generates public-private key pair and sends its secret data encrypted with public key. The other party cannot learn the data due to unavailability of private key. Similarly, the other party uses a random number to send its encrypted data to originator party. The originator cannot learn the data due to use of the random number. But the decryption results in the difference of data of both the parties. If this difference is zero the data are equal. Thus, the result of equality of both the data is known without disclosing data to one another. The proposed protocol is suitable for semi-honest adversaries. The future work suggests devising a multiparty protocol for malicious adversaries

REFERENCES

1. M. Kantarcioglu and O. Kardes, "Privacy-preserving data mining in the malicious model" in the International Journal of Information and Computer Security, Vol. 2 Issue 4, Jan 2009, pp.353 – 375.
2. Paillier, P., "Public-key cryptosystems based on composite degree residuosity classes," EUROCRYPT'99 , Prague, Czech Republic, pp.223–238, May2- 6, 1999.
3. http://en.wikipedia.org/wiki/Secure_multi-party_computation
4. C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for Privacy-Preserving Distributed Data Mining," J. SIGKDD Explorations, Newsletter, vol.4, no.2, ACM Press, pp. 28-34, Dec. 2002.
5. A. C. Yao, "protocol for secure computations," in proceedings of the 23rd annual IEEE symposium on foundation of computer science, pp. 160-164, Nov.1982.

AUTHORS PROFILE

Rashid Sheikh has received B.E. degree in Electronics and Telecommunication Engineering from Shri Govindram Seksaria Institute of Technology and Science, Indore, India in 1994 and M.Tech. degree in Computer Science and Engg. From RGPV Bhopal, India in 2010. He is pursuing PhD on "Design of Secure Multiparty Computation Protocols for Privacy Preservation". He has 25 years of teaching experience. He is the program committee member of international conferences WOCN2012 and CONSEG2012. His subjects of interest include Computer Architecture, Computer Networking, Operating Systems, Network Security and Assembly Language Programming. He has published nine research papers in International Conferences and Journals. His research areas are Secure Multiparty Computation. He has authored of three books on Computer Organization and Architecture. Currently he is working as Associate Professor at Acropolis Institute of Technology and Research, Indore, India.

Dr. Durgesh Kumar Mishra has received M.Tech. degree in Computer Science in 1994 and PhD degree in Computer Engineering in 2008 from DAVV, Indore. Presently, he is Professor (CSE) and Director at Shri Aurobindo Institute of Technology, Indore, India. He has 28 years of teaching and 15 years of research experience. He has published more than 90 papers in refereed international/national journals and conferences like IEEE, ACM conferences. He has organized many such conferences like WOCN, CONSEG and CSIBIG as conference General Chair and editor. He is a Senior Member of IEEE and held many positions like Chairman, IEEE MP-Subsection, and Chairman IEEE Computer Society Bombay Chapter, Chairman CSI Division IV Communications. He has delivered invited talk in Taiwan, Bangladesh, Singapore, Nepal, USA, UK and France. He is the author of a book "Database Management Systems". He was a consultant to sales tax and labour department of government of Madhya Pradesh, India. He has been awarded by CSI with "Paper Presenter award at International Level". He also visited MIT Boston and presented his talk on Security and Privacy, and chaired a panel on "Digital Monozukuri" at "Norbert Winner in 21st century".