A NOVEL SECURE COSINE SIMILARITY COMPUTATION SCHEME WITH MALICIOUS ADVERSARIES

Dexin Yang[1],Chunjing Lin[2] and Bo Yang[3]

[1]Department of Information,Guangzhou City Polytechnic,Guangzhou,China,510405
[2]Department of Information, Guangdong Baiyun Institute, Guangzhou,China,510460
[3]School of Computer Science, Shaanxi Normal University, 710062,
byang@snnu.edu.cn

ABSTRACT

Similarity coefficients play an important role in many aspects.Recently,several schemes were proposed,but these schemes aimed to compute the similarity coefficients of binary data. In this paper, a novel scheme which can compute the coefficients of integer is proposed. To the best knowledge of us,this is the first scheme which canesist malicious adversaries attack.

KEYWORDS

Similarity coefficients,Distributed EIGamal encryption,Zero-knowledge proof,Secure two-party computation

1. INTRODUCTION

Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. The cosine of 0 is 1, and less than 1 for any other angle; the lowest value of the cosine is -1. The cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction. Many application domains need this parameter to analyze data, such as privacy-preserving data mining, biometric matching etc.
The functionality of the privacy-preserving cosine similarity for integer data

(Denoted by$\mathcal{F}_{CC}$)can be described as follows. Consider P_1 has a vector $\mathbf{a} = (a_1, a_2, ..., a_n)$, P_2 has a vector$\mathbf{b} = (b_1, b_2, ..., b_n)$, where $a_i, b_i \in Z_p$. After the computation P_1 gets the result the cosine correlative coefficient SC and P_2 gets nothing.

Related works Secure two-party computation allows two parties to jointly compute some functions with their private inputs, while preserving the privacy of two parties private inputs. Research on the general functionality of secure computation was first proposed in [1]in the semi-honest model. Lately, Goldreich[2], Malkhi[3],Lindell and Pinkas[4,5]extended in the presence of malicious adversaries.

Even though the general solution of secure multiparty computations has given by Goldreich [6]. However, these general solutions are inefficient for practical uses, because these protocols were constructed based on the boolean circuit or the arithmetic circuit of the functionality. When the circuit of the functionality became complex enough, the complexity of this protocol will be too

DOI : 10.5121/ijnsa.2013.5213

lower to tolerate. Till now, the protocol which can resist to the attacks of malicious adversaries were the focus works of cryptographers. Therefore, it is necessary to construct the protocol which can compute cosine correlative coefficient of two vectors in the malicious model.

Kikuchi,Hiroaki et al.[7] gave the first protocol to compute two vectors cosine correlative coefficient based on zero knowledge proof of range and applied this protocol to biometric authentication. This protocol is based on zero-knowledge proofs and Fujisaki-Okamoto commitments[8]. Recently, K.S.Wong et al.[9] proposed a new protocol which can compute the similarity coefficient of two binary vectors in the presence of malicious adversaries. Later, Bo zhang et al.[10] pointed out this scheme is not secure, and another scheme which can overcome the shortage of Wong's scheme is proposed.

Our results In this paper, a new protocol which can compute the cosine correlative coefficient is proposed. Our protocol can resist the attacks of malicious adversaries, and we give the standard simulation-based security proof.

Our main technical tools include distributed EIGamal encryption[11] and zero-knowledge proofs of knowledge. The main property of distributed EIGamal encryption is that the parties must cooperate while in decrypting stage because each party has partial decrypt key.

2.PRELIMINARIES

2.1 Cosine Correlative Similarity

Let $\mathbf{a} = (a_1, a_2, ..., a_n), \mathbf{b} = (b_1, b_2, ..., b_n)$ be two $n-$ dimensional integer vectors.
We consider the cosine similarity between $\mathbf{a}$ and $\mathbf{b}$,which will be evaluated in privacy-preserving in later section.

Definition 1 A cosine correlation is a similarity between $\mathbf{a}$ and $\mathbf{b}$ defined as

$$cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|} = \frac{a_1 b_1 + ... + a_n b_n}{\sqrt{a_1^2 + ... + a_n^2}\sqrt{b_1^2 + ... + b_n^2}}$$

For normalization $\mathbf{a}, \mathbf{b}(\|\mathbf{a}\| = \mathbf{1}, \|\mathbf{b}| = \mathbf{1}$,the cosine correlation can be simplified as $cos(\mathbf{a}, \mathbf{b}) \quad \mathbf{a} \cdot \mathbf{b} \quad \mathbf{a_1 b_1} + ... + \mathbf{a_n b_n}$ where $\|\mathbf{a}\|$ is a norm of $\mathbf{a}$.

The proposed scheme in this paper is focus on computing the cosine correlation coefficient of two normalization vectors$\mathbf{a}$ and $\mathbf{b}$.

2.2 Distributed ElGamal Encryption

ElGamal encryption [11] is a probabilistic and homomorphic public-key crypto system. Let p and q be two large primes such that q divides$p - 1$. G_q denotes Z_p^* unique multiplicative subgroup of order q. All computations in the remainder of this paper are modulo$p$ unless otherwise noted. The private key is $x \in Z_q$, and the public key is $y - g^x$ $(g \in G_q$ is a generator). A message $m \in G_q$ is encrypted by computing the ciphertext tuple $(\alpha, \beta) \quad (my^r, g^r)$ where r is an arbitrary random number in Z_q, chosen by the encrypter.

A message is decrypted by computing

$$\frac{\alpha}{\beta^x} = \frac{my^r}{(g^r)^x} = m$$

ElGamal is homomorphic , as the component-wise product of two ciphertexts

$$(\alpha\alpha', \beta\beta') = (mm'y^{r-r'}, g^{r+r'})$$

represents an encryption of the plaintexts product mm'.

A distributed EIGamal Encryption system[12]is a public-key cryptosysytem which
key generation algorithm [13]and decryption algorithm is as follows:

Distributes key generation: Each participant chooses x_i at random and publishes $y_i = g^{x_i}$ along with a zero-knowledge proof of knowledge of y_i's discrete logarithm. The public key is $y = \Pi_{i-1}^{n} y_i$, the private key is $x = \Sigma_{i-1}^{n} x_i$. This requires n multiplications, but the computational cost of multiplications is usually negligible in contrast to exponentiations.

Distributed decryption: Given an encrypted message (α, β), each participant

publishes $\beta_i = \beta^{x_i}$ and proves its correctness by showing the equality of logarithms of y_i and β_i. The plaintext can be derived by computing $\frac{\alpha}{\Pi_{i-1}^{n} \beta_i}$. Like key generation, decryption can be performed in a constant number of rounds, requiring n multiplications and one exponentiation.

Same as Bo zhang et al.[10],we also use an additively homomorphic variation of EIGamal Encryption with distributed decryption over a group $\mathbb{G}_q$ in which DDH is hard, i.e,$E_{pk}(m, r) = (g^r, g^m h^r)$.

2.3 Zero Knowledge Proof

In order to obtain security against malicious adversaries, the participants are required to prove the correctness of each protocol step. Zero knowledge proof is a primitive in cryptography.
In fact, the proposed protocols can be proven correct by only using Σ–protocols. A Σ–protocol is a three move interaction protocol. In this paper, there are four Σ–protocols are used as follows.

We denote these associated functionalities by $\mathcal{F}_{DL}, \mathcal{F}_{EqDL}, \mathcal{F}_{KeyGen}, \mathcal{F}_{IsCipher}$.Next, we simply describe the associated zero-knowledge protocols:$\pi_{DL}, \pi_{EqDL}, \pi_{KeyGen}, \pi_{IsCipher}$.

π_{DL}. The prover can prove to the verifier that he knows the knowledge of the
solution x to a discrete logarithm.

$$R_{DL} = \{((G_q, q, g, h), x) | h = g^x\}$$

π_{EqDL}. The prover can prove to the verifier that the solutions of two discrete logarithm problems are equal.

$$R_{EqDL} = \{((G_q, q, g, g_1, g_2, g_3), x) | g_1 = g^x \wedge g_3 = g_2^x\}.$$

π_{KeyGen}. The prover can prove to the verifier that the generation of EIGamal encryption is valid.

$$R_{KeyGen} = \{((G_q, q, g), s_1, s_2) | h = g^{s_1 + s_2}\}$$

$\pi_{IsCipher}$. The prover can prove to the verifier that the ciphertext of EIGamal encryption is valid

$$R_{IsCipher} = \{(G_q, q, g, h), m) | (c_1 = g^r \wedge c_2 = g^m h^r)\}$$

3.THE PROPOSED SCHEME

In this section,we give out the protocol(Π_{SC}) which computes the coefficient of two integer vectors in the presence of malicious adversaries. The ideal functionality of coefficient $\mathcal{F}_{SC}$ is as follows:

$$((a_1, a_2, ..., a_n), (b_1, b_2, ..., b_n)) \mapsto (SC, \lambda)$$

where λ denotes P_2 gets nothing after the protocol execution,SC denotes the cosine coefficient between two vectors $\mathbf{a}, \mathbf{b}$. In the ideal model,P_1 sends his private input $\mathbf{a}$ to the third trusted party(TTP), similarly P_2 sends his private input $\mathbf{a}$ to TTP. Finally, TTP sends SC back to P_1, and nothing to P_2.

The building blocks of our protocol include distributed EIGamal encryption and zero-knowledge proofs.The reason we choose distributed EIGamal encryption rather than original EIGamal encryption is the distributed EIGamal encryption is less complexity in protocol.

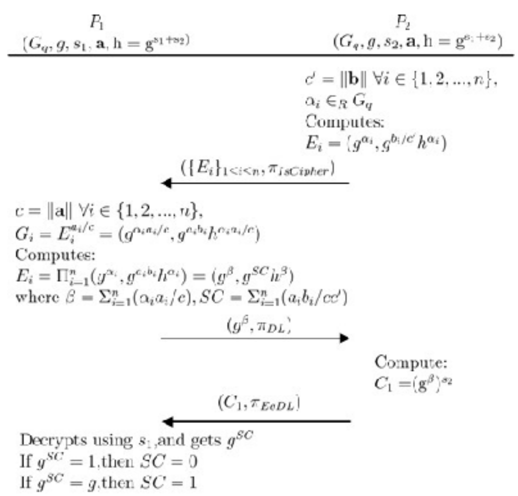

Fig 1 The proposed scheme

The protocol(Π_{SC})(Fig 1) is as follows:

-**Inputs:** The input of P_1 is a n dimensional integer vector $\mathbf{a} = (a_1, a_2, ..., a_n)$.Similarly, P_2's input is $\mathbf{b} = (b_1, b_2, ..., b_n)$.

-Auxiliary Inputs: Both parties have the security parameter 1^{κ}.

-The protocol:

1. P_1, P_2 engage in the protocol $\pi_{KeyGen}(1^{\kappa}, 1^{\kappa})$ to generate the public key $pk = (G_q, g, h = g^{s_1 - s_2})$,and the private key$(s_1, s_2)$,shared by P_1, P_2 respectively.

2. P_2 computes: $c' = \|\mathbf{b}\|, E_i = (g^{\alpha_i}, g^{b_i/c'} h^{\alpha_i}), i \in \{1, 2, ..., n\}$,and sends E_i to P_1.The parties run the zero-knowledge proof of knowledge $\pi_{iscipher}$,allowing P_1 to verify that the ciphertext E_i is valid.

3.Upon receiving the E_i from P_2,the P_1 computes:

$c = \|\mathbf{a}\|, \forall \mathbf{a_i}, \mathbf{i} \in \{\mathbf{1}, \mathbf{2}, ..., \mathbf{n}\}$, $\quad G_i = E_i^{a_i/c} = (g^{\alpha_i a_i/c}, g^{a_i b_i/c}) h^{\alpha_i a_i/c})$,
$E_{SC} = \Pi_{i-1}^{n} G_i = (g^{\Sigma_{i-1}^{n}(\alpha_i a_i/c)}, g^{\Sigma_{i-1}^{n}(a_i b_i/cc')} h^{\Sigma_{i-1}^{n}(\alpha_i a_i/c)})) = (g^{\beta}, g^{SC} h^{\beta})$

where $\beta = \Sigma_{i-1}^{n}(\alpha_i a_i/c), SC = \Sigma_{i-1}^{n}(a_i b_i/cc')$.and sends E_{SC} to party P_2.The parties run the zero-knowledge proof of knowledge $\pi_{iscipher}$,allowing P_2 to verify that the ciphertext E_{SC} is valid.

4.Upon receiving the (g^{β}, π_{DL}) from P_1 , P_2 computes C_1 using his private key s_2 as: $C_1 = (g^{\beta})^{s_2}$,and send C_1 to P_1.The parties run the zero-knowledge proof of knowledge π_{EqDL},allowing P_1 to verify that C_1 is valid.

5.Upon receiving the C_1 from P_2, P_1 decrypts and obtains g^{SC}, where SC is the cosine coefficient of the two vectors $\mathbf{a}, \mathbf{b}$.

At last, P_1 evaluates SC as follows.

(a)If $g^{SC} = 1$,then $SC = 0$;

(b)If $g^{SC} = g$,then $SC = 1$.

4. SECURITY ANALYSIS

Theorem 1 Assume that $\pi_{DL}, \pi_{EqDL}, \pi_{KeyGen}, \pi_{IsCipher}$ are as described in section 2 and that (Gen, E, D) is the EIGamal scheme. The Π_{SC} correctly evaluates the consine coefficient of two n -dimension variables in the presence of malicious adversaries.
The proof of this theorem 1 is obviously.

Theorem 2 Assume that $\pi_{DL}, \pi_{EqDL}, \pi_{KeyGen}, \pi_{IsCipher}$ are as described in section 2 and that (Gen, E, D) is the EIGamal scheme. The Π_{SC} securely evaluates the consine coefficient of two n -dimension variables in the presence of malicious adversaries.

Proof:We prove this theorem in the hybrid model,where a third trusted party is introduced to compute the ideal functionality $\mathcal{F}_{DL};\mathcal{F}_{EqDL};\mathcal{F}_{KeyGen};\mathcal{F}_{IsCipher}$. As usual, we analyze two cases as P_1 is corrupted and P_2 is corrupted separately.

P_1 **is corrupted**. Assume that P_1 is corrupted by adversary $\mathcal{A}$ with the auxiliary input z in the real model. we construct a simulator $\mathcal{S}$,who runs in the ideal model with the third trusted party computing the functionality F_{D_C}.$\mathcal{S}$ works as follows.

1.$\mathcal{S}$ is given $\mathcal{A}$'s input and auxiliary input, and invokes $\mathcal{A}$ on these values.

2. $\mathcal{S}$ first emulates the trusted party for π_{KeyGen} as follows. It first two random elements $s_1, s_2 \in \mathbb{Z}_q$, and hands $\mathcal{A}$ $s_1$and the public key $(\mathbb{G}_u, q, g, h = g^{s_1 - s_2})$.

3.$\mathcal{S}$ receives from P_2,n encryptions and P_2's input for the trusted party for $F_{iscipher}$, then define $\mathcal{A}$'s inputs as $\mathbf{b}$.

4.Then $\mathcal{S}$ sends $\mathbf{b}$ to the trusted party to compute F_{SC} to complete the simulation in the ideal model. Let I_{D_C} be the returned value from the trusted party.

5.Next $\mathcal{S}$ randomly chooses $\mathbf{a}' = (\mathbf{a}'_1, \mathbf{a}'_2, ..., \mathbf{a}'_n)$ conditioned on that the cosine coefficient equals to I_{D_C}. $\mathcal{S}$ completes the execution as the honest party P_2 would on inputs $\mathbf{a}'$.

6.If at any step, $\mathcal{A}$ sends an invalid message, $\mathcal{S}$ aborts sends $\perp$ to the trusted party for F_{D_C}. Otherwise, it outputs whatever $\mathcal{S}$ does.

The difference between the above simulation and the real hybrid model is that $\mathcal{S}$ who does not have the real

P_1's input $\mathbf{a}$,simulates following steps with the randomly chosen $\mathbf{a}$ under the condition that the output of them are the same.The computationally distinguishability of them can be deduced from the sematic security of EIGamal encryption. In other words, if $\mathcal{A}$ can distinguish the simulation from the real execution,we can construct a distinguisher $\mathcal{D}$ to attack the semantic security of EIGamal encryption.

P_2 **is corrupted**. The proof of this part is similar with above.We construct a simulator $\mathcal{S}$ in the ideal model, based on the real adversary $\mathcal{A}$ in the real model. $\mathcal{S}$ works as follows.
1.$\mathcal{S}$ is given $\mathcal{A}$'s input and auxiliary input, and invokes $\mathcal{A}$ on these values.
2. $\mathcal{S}$ first emulates the trusted party for π_{KeyGen} as follows. It first two random elements $s_1, s_2 \in \mathbb{Z}_q$, and hands$\mathcal{A}$ s_1 and the public key$(\mathbb{G}_u, q, g, h = g^{s_1 - s_2})$.
3.$\mathcal{S}$ randomly chooses$\mathbf{b}' = (\mathbf{b}'_1, \mathbf{b}'_2, ..., \mathbf{b}'_n)$, then encrypts them using the public key.
4.Next,$\mathcal{S}$ sends the ciphertexts to $\mathcal{A}$, and proves to $\mathcal{A}$ that all the ciphertexts is valid using $\pi_{iscipher}$.
5.$\mathcal{S}$ receives from $\mathcal{A}$ n ciphertexts and $\mathcal{A}$'s input to the trusted party for$F_{iscipher}$, then defines $\mathcal{A}$'s inputs as$\mathbf{b}'$.
6.The $\mathcal{S}$ completes the next step as the honest P_1.
7.If at any step, $\mathcal{A}$ sends an invalid message, $\mathcal{S}$ aborts sends $\perp$ to the trusted party for F_{D_C}.Otherwise $\mathcal{S}$ sends $\mathbf{b}'$ to the trusted party computing F_{SC},and outputs whatever $\mathcal{S}$ does.

Similar to the case P_1 is corrupted, the difference between the simulation and the real model is that $\mathcal{S}$ uses $\mathbf{b}'$ as P_2's input. However, $\mathbf{b}'$ is encrypted by the public key of a semantic security EIGamal encryption. Same as the above,the analysis of this simulation distribution can be assured by the definition of zero-knowledge proof and semantic security of a public-key encryption.

In summary, we complete the proof of Π_{SC} in the presence of malicious adversaries.

5. CONCLUSION

Similarity coefficients (also known as coefficients of association) are important measurement techniques used to quantify the extent to which objects resemble one another. There are various similarity coefficients which can be used in different fields. Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. In this paper, a new scheme which can compute the cosine correlative of two integer vectors in the presence of malicious adversaries.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant This work is supported by the National Natural Science Foundation of China under Grant 61173164 and 61272436, and the Natural Science Foundation of Guangdong Province under Grants 10351806001000000.

REFERENCES

[1] Andrew Chi-Chih Yao. How to generate and exchange secrets. In Proceedings of the 27th Annual Symposium on Foundations of Computer Science, SFCS '86, pages162-167, Washington, DC, USA, 1986. IEEE Computer Society.

[2] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In Proceedings of the nineteenth annual ACM symposium on Theory of computing,STOC '87, pages 218-229, New York, NY, USA, 1987. ACM.

[3] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay—a secure two-party computation system. In Proceedings of the 13th conference on USENIX Security Symposium - Volume 13, SSYM'04, pages 20-20, Berkeley, CA,USA, 2004. USENIX Association.

[4] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In Proceedings of the 26th annual international conference on Advances in Cryptology, EUROCRYPT '07, pages 52-78, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Yehuda Lindell and Benny Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In Proceedings of the 8th conference on Theory of cryptography, TCC'11, pages 329-346, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] O.Goldreich. Secure multi-party computation (working draft),1998.http://citeseer.ist.psu.edu/goldreich98secure.html.

[7] Hiroaki Kikuchi, Kei Nagai, Wakaha Ogata, and Masakatsu Nishigaki. Privacy-preserving similarity evaluation and application to remote biometrics authentication. Soft Comput., 14(5):529-536, December 2009.

[8] S.Wong and M.H.Kim. Privacy-preserving similarity coe_cents for binary data. Computers and Mathematics with Applications, 2012.http://dx.doi.org/10.1016/j.camwa.2012.02.028.

[9] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '97, pages 16-30,London, UK, UK, 1997. Springer-Verlag.

[10] Bo Zhang and Fangguo Zhang. Secure similarity coefficients computation with malicious adversaries. Cryptology ePrint Archive, Report 2012/202, 2012.http://eprint.iacr.org/.

[11] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of CRYPTO 84 on Advances in cryptology,pages 10-18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[12] Felix Brandt. Efficient cryptographic protocol design based on distributed el gamal encryption. In Proceedings of the 8th international conference on Information Security and Cryptology, ICISC'05, pages 32-47, Berlin, Heidelberg, 2006. Springer-Verlag.
[13] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '91, pages 129-140, London, UK, UK,1992. Springer-Verlag.

Authors

Dexin Yang received his Ph.D. degree at South China Agricultural University. Now he is a lecturer of Guangzhou City Polytechnic. His main research topics are cryptographyand information security.

Bo Yang received his Ph.D. degree at Xidian University in China. Now he is a professor of Shaanxi Normal University. His main research topics are cryptography and information security.

Chunjing Lin received his master degree at Xidian University in China. Now he is a professor of Guangdong Baiyun Institute. His main research topics are embedded system and information processing