Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Algorithm for SMC with Analysis of Malicious Conduct

Samiksha Shukla*
Computer Science & Engineering
Christ University, Bangalore, India

G Sadashivappa
Telecommunication Engineering
R V College of Engineering, Bangalore, India

Abstract— The aim of the proposed study of is to implement secure multi-party computation protocol in anonymized and randomization environment where parties have access to trusted third party (TTP) who (1) doesn't contribute any input for computation (2) doesn't get actual input of the individual parties (3) has vast number of resources and randomization function is known to get the actual data for computation of final result. In this environment we are interested in designing a protocol that outsources computation to TTP. It is proposed that the protocol is very efficient (in terms of computation and privacy) for the parties than the other existing protocols. The solution incorporates protocol with TTP, where there is a possible security threat. As the protocol works on randomization and packetization it ensures following: (1) Confidentiality (Anonymity) (2) Security (3) Privacy (Data).

Keywords— Randomization, Secure multi-party computation (SMC), Packetization, trusted third party (TTP), Privacy, Anonymization

I. INTRODUCTION

Joint computation is often required in various real life scenarios that depend on the private input of numerous parties but they can not afford to disclose their input to each other. It can occur during medical analysis, voting, auction, data mining etc. SMC allows parties to securely perform cooperative computations over their confidential data. Generally SMC ensures (1) parties will not learn anything than the output (2) computation is done correctly. [1,3,4] demonstrated, any functionality can be computed securely using 'SMC'. Various works are going on to improve the security by enhancing number of malicious parties, reducing number computation and computation complexities. Most of the work is based on homogeneous computing environment. In practice computation may not be in such environment. Cloud computing is one of the application where heterogeneous setting is wisely used. In cloud computing a computationally powerful service provider provide access to clients. In this study the issue of secure computation in an environment where along with the parties, there is a trusted third party (TTP) that (1) performs the joint computation over private input of parties (2) Doesn't contribute any input to computations (3) has numerous resources for computations. Above mentioned setting is considered for the proposed protocol design. It minimizes the computation of parties at the expense of the trusted third party (TTP) (as maximum work need to be done by TTP). [6,5,8] proposes new advances in SMC that can become practical. One remarkable exception in secure multi-party computation (SMC) is the work of [10] they considered a setting with anonymization to present the protocol which permit reducing the client's work at the expense of trusted third party (TTP).

II. RELATED WORK

With SMC, a number of parties can cooperatively perform some global function on their private data without any loss of data privacy. It provides support for end-to-end secure multiparty protocol development. Let parties P_1, P_2... P_n be n parties (organizations or individuals) who wish to perform a cooperative computation C_i on their private data. Since, computation is to be carried out on private data, it is key constraint that this private data should not be available to any other party, i.e. if $D_1, \ldots, D_n$ be the data corresponding to n parties and let D_i be data corresponding to i^{th} party, then it is required for computation that, D_i should not be accessible to any D_j where $i \neq j$ *and* $j=1, 2 \ldots n$. Therefore, each party only gets the final results of cooperative computation without being aware of inputs involved and the computations made. Many models have been proposed in the literature for the SMC problems. Generally, two model prototypes are popular:

Ideal Model Prototype of SMC
Real Model Prototype of SMC

Ideal model considers an uncorrupted Trusted Third Party (UTTP) among participating parties. Parties send their private data inputs to the UTTP who is supposed to perform computation on behalf of these parties. The UTTP is assumed to be honest in the sense that it never reveals the private data of one party to others. The UTTP, after valuation of common function sends the value of the outcome to all the participating parties. Only computations result is known to all the parties. Thus the privacy of the input is preserved. In this model, if a few parties behave maliciously then the result of the computation may be incorrect because the party may supply invalid input to the UTTP but the individual parties privacy will be preserved. If the UTTP turns corrupt, the privacy may be destroyed. The ideal model of SMC for two

parties is shown in fig. 1. Same can be extended for multiple parties. Participating parties provide their private data inputs $x_1, x_2 \ldots x_n$ to the UTTP. The UTTP then evaluates some function $f(x_1, x_2 \ldots x_n)$ and sends back this value to all the participating parties. In practical scenario the role of UTTP is played by some government or private organization which works as a service provider for this computation. The ideal model of SMC is costly due to the cost of working of the UTTP. One more drawback of this model is that the trustworthiness of the UTTP is very important. When UTTP turns corrupt whole idea of the SMC becomes insignificant. But today this model is frequently used due to easy implementation and use of tools that prevent the UTTP to become malicious.

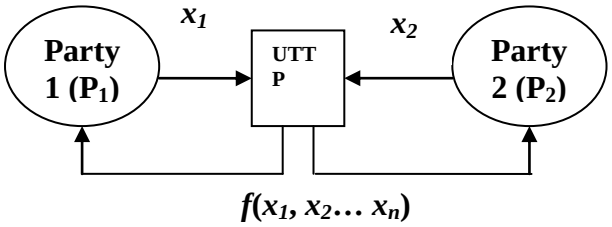

Fig. 1 Ideal Model of SMC

In real model no external party is assumed which can be trusted. In this model participating parties agree on a protocol which is to be run among them in order to preserve privacy and maintain correctness result. This model for two parties is shown in fig. 2. Parties do not contribute definite inputs with each other. The values sent by parties are some function of their private data. What exists between parties is a theoretical computation mechanism. The real model of SMC is said to be secure if an adversary can carry out some attack which is also possible in the ideal model of SMC. An adversary is a party with malicious purpose. An adversary can be static or adaptive in nature. A static adversary is malicious in nature prior to the execution of the protocol. An adaptive adversary turns malicious during the execution of the protocol. A semi-honest adversary follows the protocol but tries to learn something other than the output of the computation. A corrupt or malicious adversary is that which does not follow the steps of the protocol and tries to learn some information other than the result.

Fig. 2 Real Model of SMC

Researchers have classified several general SMC problems that include privacy- preserving data mining, privacy-preserving database query, privacy-preserving statistical analysis, privacy-preserving geometric computation, secure biometric computation etc. [1] [14]. The initial concept of secure multi-party computation (SMC) has been brought in notice by Yao in [1] and then in [5] Maurer extended it by defining different type of security and the applications of secure multi-party computation (SMC). The first solution for this problem has been proposed by Yao in the form of two party computations for semi-honest party. Its extensions to malicious party were given by Lindell [12]. Lindell et. al. defined the problem as, there exist two parties with their individual databases and want to perform data mining operation on union of their database. Similar problem was addressed by agarwal et. al. Both researches came out with different solutions agarwal solved it using data perturbation while Lindell considered secure multi-party computation (SMC) techniques. Few existing protocols are in the form of 1-out of-N oblivious transfer, circuit evaluation, fully homomorphic encryption which has been used by [13] [14]. The work more closely related to the proposed protocol has been presented by Mishra and Chandwani [11]. They considered a setting which includes multiple parties with private input and a trusted third party without any contributory input. In this setting parties send data to randomly chosen anonymizer. And then anonymizers forward the data to the trusted third party (TTP). In this protocol there is a possibility of misconduct as data is sent as it is to the anonymizer and if anonymizer becomes malicious and collide with any one party then protocol may break. The proposed protocol settings slightly vary from the work proposed in [10]. It is illustrated that it is secure against malicious trusted third party (TTP) and anonymizer.

III. OVERVIEW OF PROPOSED PROTOCOL

The protocol is based on Ideal model prototype of secure multi-party computations (SMC). In proposed protocol parties interact only to generate random number with the help of coin tossing algorithm. It will be performed exactly once to generate a random number (r) on which all the parties are mutually agree. Random number is used by randomization function for data privacy. Randomization function random (o, x_i, r) would be decided by one arbitrarily chosen party and shared among all other parties and trusted third party (TTP). Once it is received by all the parties, parties will use randomization function to hide the actual input then break it in fixed number of packet and distribute among various anonymizers. Anonymizers then forward the packet to trusted third party (TTP). Trusted third party (TTP) derandomizes the data with the help of randomization function to perform the joint computation.

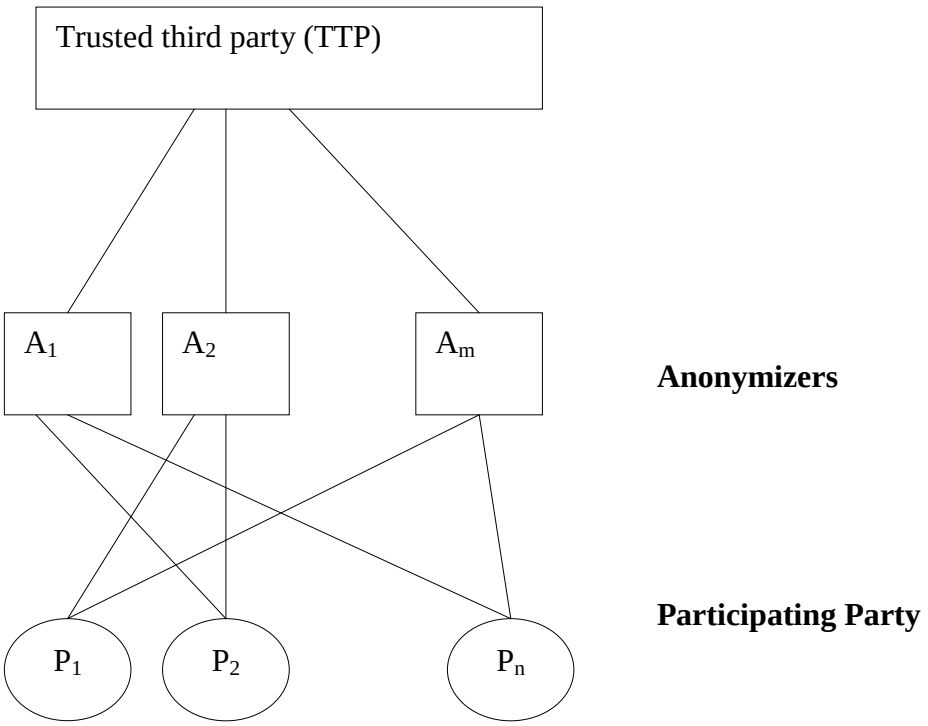

Fig. 3 JCRA Architecture

A. Protocol (Proposed)

Inputs: (P_1, P_2,…..,P_n) are parties with (x_1, x_2, …,x_n) inputs.
TTP has no input, all the parties and TTP knows random number r and randomization function random(o, x_i, r).
Outputs: All the parties and TTP learn $f(x_1, x_2, \ldots, x_n)$.
Step 1: All the 'n' parties jointly runs coin tossing algorithm to generate random number r, as a result all the parties learn r.
Step 2: One Randomly selected party generate randomization function random(o, x_i, r) and share it with 'n' parties and trusted third party (TTP) via virtual link.
Step 3: All the 'n' parties then computes
$R(x_1)$ = random (o, x_1, r)
$R(x_2)$ = random (o, x_2, r)
$R(x_n)$ = random (o, x_n, r)
Step 4: Each party divides the randomized data in fixed number of packet and forward it to randomly selected anonymizer.
Step 5: Anonymizers then forward it to trusted third party (TTP).
Step 6: After receiving complete data of all the parties trusted third party (TTP) derandomize the data and evaluates joint computation function f(D) to announce the desired result.

B. Algorithm (Proposed)

Joint Computation with Randomization and Anonymization (JCRA)
Assumptions: 1. Number of packets are same for all the parties (t_p).
2. Anonymizers are only to forward data.
3. TTP computes correctly.
4. All the parties are giving valid input.
Inputs: ($x_1, x_2, \ldots, x_n$) parties input for each party respectively, random(o, x_i, r), random number (r), Number of packets (t_p).

Output: f(D)
Variable list: n - Number of parties.
t_p - Number of packets.
A_r - Randomly selected anonymizer.
T_{ann} - Total number of anonymizer.
S_D - Combine input of all the parties.
C_{tp} - Total packet count at TTP.

E_{tp} - Expected number of packet at TTP.
Max_limit_anonymizer- Maximum number of packet an anonymizer can receive.

//Phase 1: Randomization

- Parties Jointly Execute coin tossing to generate r;
- P_i generates random(o, x_i, r) and share with TTP and parites;
- for (i = 1 to n) do

begin
a) $R(x_i)$ = random(o, x_i, r); // randomize the individual data

// Phase 2: Packetization

b) divide $R(x_i)$ in t_p parts. // (P_iK_1, P_iK_2… P_iK_{tp})
for (j=1 to t_p) do
begin
c) randomly select one anonymizer A_r;
d) if (count(A_r)<Max_limit_anonymizer)then
begin
send P_iK_j to A_r;
increase the count(A_r) by 1;
end;
else
e) chose another anonymizer and repeat step 3(c);
end;
end;

//Phase 3: Data Collection at TTP

for (j = 1 to $T_{ann)}$ do
begin
for (i = 1 to packet in anonymizer) do
begin
redirect packet to TTP;
TTP will append packet to S_D;

$$S_D = \sum_{i=1}^{n} Di + r$$

C_{tp} increased by 1; // increase total packet count by 1.
end;
end;

//Phase 4: Data Verification and Computation

begin
E_{tp}=n*tp;
If (C_{tp} = E_{tp}) then
Compute *f*(D) by Derandomizing S_D using r.
Broardcast the result *f*(D);
else
return 'packets lost';
end;

Case 1: Malicious conduct by certain parties- All the parties will have following information: i) Random number r and randomization function, ii) total number of packets of individual party iii) Parties own data. This information is not sufficient to get any third parties data even if some parties collide.

Case 2: **Malicious conduct by Anonymizers**- In the proposed protocol anonymizer receives randomized packets P_iK_{tp} where tp is total number of packet of individual parities. In this case even if few anonymizers collude they will not get any information until they get randomization function and random number 'r' as data is randomized.

In case, if anonymizer gets randomization function and random number 'r' then the probabilistic representation of breaking protocol will be:

$$\Pr(k, m) = \frac{1}{m^k} \quad (1)$$

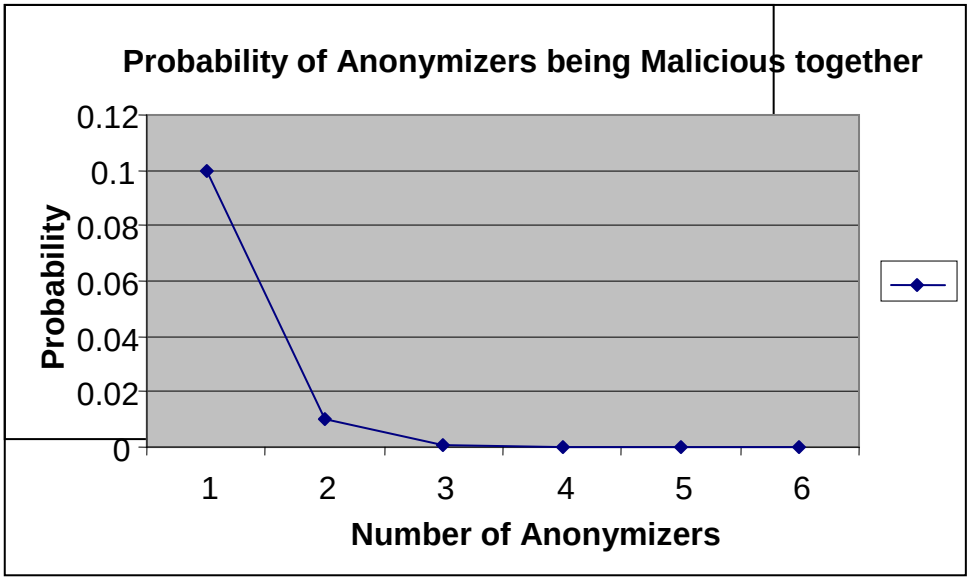

Fig. 4 Probability of breaking when anonymizers collude

It is clear from the figure 4, if number of anonymizers is four or more the probability of breaking the protocol is insignificance.

Case 3: Joint Malicious conduct by anonymizers and parties: In the proposed protocol number of parties are 'n' and number of anonymizers are 'm'. Following information is known to parties: i) Random number r and randomization function, ii) total number of packets of individual party iii) Parties own data. Anonymizers receive randomized data packets from different parties, as the parties packets are distributed among m anonymizers probability of getting any parties data is negligible until they collide.

If '*k*' anonymizers collide with '1' party out of 'n' then probability will becomes:

$$\Pr(1, k) = \frac{1}{n} \times \frac{1}{m^k} \quad (2)$$

If '*k*' anonymizers collide with '*l*' parties then probability will becomes:

$$\Pr(l, k) = \frac{1}{n^l} \times \frac{1}{m^k} \quad (3)$$

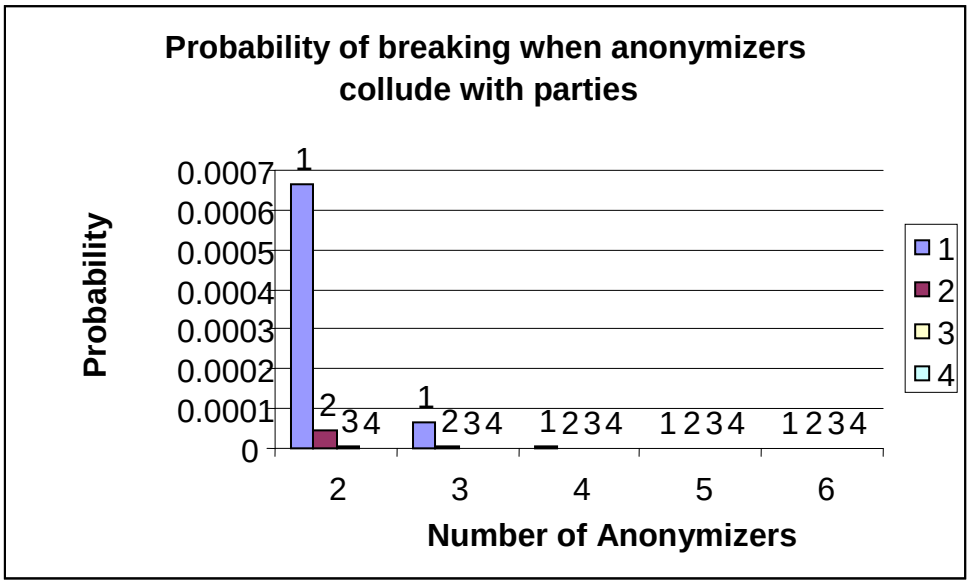

Fig. 5 Probability of breaking when anonymizers collude with parties

Case 4: Joint Malicious Conduct by TTP and Anonymizers: In the proposed protocol single TTP is considered for joint computations and number of anonymizers are 'm'. Following information is known to TTP: i) Random number r

and randomization function, ii) total number of packets of all the parties. In case if TTP behave maliciously then probability of breaking the protocol will be:

$$\Pr(1, k) = \frac{1}{m^{k+1}} \quad (4)$$

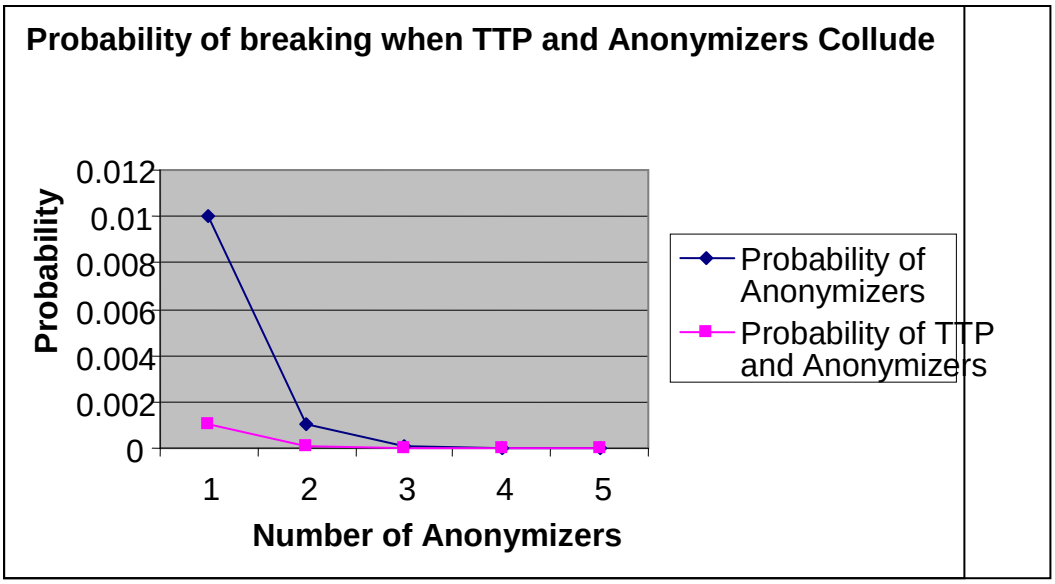

Fig. 6 Probability of breaking when anonymizers collude with TTP

IV. CONCLUSIONS

The requirement of privacy preservation is gradually rising in the world due to the extensive use of online services provided by untrusted parties having access to user's private information such as their personal inclinations or medical details. In this paper, it has been shown how proposed protocol can help to attain the twofold objective of achieving privacy and security of parties' private input. The proposed protocol deals with the layered architecture where in coin tossing algorithm for randomization and anonymization are used. While the proposed protocol is relatively efficient therefore focuses on promising paths for real world applications.

ACKNOWLEDGMENT

Authors wish to acknowledge Dr. Durgesh Mishra, Professor and Head (CSE) Sri Aurobindo Institute of Technology and Research, Indore and Dr. Ravi Prakash G, Professor Alliance University, Bangalore for their guidance.

REFERENCES

[1] A. Yao. Protocols for secure computations. In IEEE Symposium on Foundations of Computer Science (FOCS '82), pages 160-164. IEEE Computer Society, 1982.

[2] A. Yao. How to generate and exchange secrets. In IEEE Symposium on Foundations of Computer Science (FOCS '86), pages 162-167. IEEE Computer Society, 1986.

[3] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In ACM Symposium on the Theory of Computation (STOC '87), pages 218-229, ACM, 1987.

[4] D. Chaum, C. Cr_epeau, and I. Damgard. Multiparty unconditionally secure protocols. In ACM symposium on Theory of computing (STOC '88), pages 11-19. ACM, 1988.

[5] Maurer, Ueli. "The role of cryptography in database security." In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 5-10. ACM, 2004.

[6] P. Bogetoft, D. Christensen, I. Damg_ard, M. Geisler, T. Jakobsen, M. Kr_igaard, J. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Secure multiparty computation goes live. In Financial Cryptography and Data Security (FC '09), pages 325-343, Springer-Verlag, 2009.

[7] Y. Lindell and B. Pinkas. An effcient 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.

[8] Y. Lindell, B. Pinkas, and N. Smart. Implementing two-party computation effciently with security against malicious adversaries. In Proceedings of the 6th international conference on Security and Cryptography for Networks (SCN '08), pages 2-20, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] B. Pinkas, T. Schneider, N. Smart, and S. Williams. Secure two-party computation is practical. In Advances in Cryptology - ASIACRYPT '09, pages 250-267. Springer-Verlag, 2009.

[10] Kamara, Seny, Payman Mohassel, and Mariana Raykova. "Outsourcing Multi-Party Computation." *IACR Cryptology ePrint Archive* 2011 (2011): 272.

[11] Mishra, Durgesh Kumar, and Manohar Chandwani. "A zero-hacking protocol for secure multiparty computation using multiple TTP." In *TENCON 2008-2008 IEEE Region 10 Conference*, pp. 1-6. IEEE, 2008..

[12] Mishra, D. K., and M. Chandwani. "Anonymity enabled secure multi-party computation for Indian BPO." In *TENCON 2007-2007 IEEE Region 10 Conference*, pp. 1-4. IEEE, 2007.

[13] Lindell, Yehuda, and Benny Pinkas. "Privacy preserving data mining." *Journal of cryptology* 15, no. 3 (2002): 177-206.

[14] Sadeghi, Ahmad-Reza, Thomas Schneider, and Marcel Winandy. "Token-based cloud computing." In *Trust and Trustworthy Computing*, pp. 417-429. Springer Berlin Heidelberg, 2010.

[15] Rane, Shantanu, Ye Wang, Stark Draper, and Prakash Ishwar. "Secure Biometrics: Concepts, Authentication Architectures and Challenges." *arXiv preprint arXiv:1305.4832* (2013).

