My First Contribution to Solve Problems Related to Size of the Payload Associated with SMCs Using Trusted Pals

Mr.S.Samson Dinakaran
Assistant Professor/Department of CS
VLB Janakiammal College of Arts & Science,
Coimbatore, Tamil Nadu, India.
samsondinakarans@gmail.com

Dr.M.Devapriya
Assistant Professor/Department of CS
Government Arts College,
Coimbatore, Tamil Nadu, India.
devapriya_gac@rediffmail.com

Abstract: **Secure Multiparty Computation(SMC) (also known as secure computation or multi-party computation/MPC) is a subfield of cryptography with the goal of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private. TrustedPals allows reducing SMC to the problem of fault-tolerant consensus between smartcards, where only process crashes and message omissions may take place. Hence, within the redesign aimed at incorporating failure detection, we investigate the problem of solving consensus in such an omission failure model augmented with failure detectors. This papers covers about the contribution to solve some of the SMC problems by considering the size of the payload is to be chosen. It is necessary to find an acceptable tradeoff between security and performance such that a message size provides better security in expense of worse performance..**

I. INTRODUCTION

Secure multi-party computation (also known as secure computation or multi-party computation/MPC) is a subfield of cryptography with the goal of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private. We present a modular redesign of Trusted Pals, a smartcard-based security framework for solving secure multiparty computation (SMC) problem also known as secure computation or multi-party computation (MPC), a subfield of cryptography. The goal of methods for secure multi-party computation is to enable parties to jointly compute a function over their inputs, while at the same time keeping these inputs private. TrustedPals allows reducing SMC to the problem of fault-tolerant consensus between smartcards, where only process crashes and message omissions may take place. Hence, within the redesign aimed at incorporating failure detection, we investigate the problem of solving consensus in such an omission failure model augmented with failure detectors. The sub-problem of MPC that has received special attention by researchers because of its close relation to many cryptographic tasks is referred to as secure two-party computation (2PC) or just as Secure function evaluation (SFE). This area of research is concerned with the question: 'Can two party computation be achieved more efficiently and under weaker security assumptions than general MPC?' We make a comparative study of several protocols for SMC and try to identify a problem to be solved and implemented. The first contribution is the size of the payload is to be chosen.

II. FIRST CONTRIBUTION

The first contribution is the size of the payload is to be chosen. It is necessary to find an acceptable tradeoff between security and performance such that a message size provides better security in expense of worse performance. So, we use an adaptive model for the tradeoff between service performance and security in service-based environments is presented. The performance and security metrics allow us to quantitatively calculate how much protection a security configuration vector can provide and how much performance will be decreased by that security configuration vector.

III. MINIMUM REQUIREMENT VALIDATION

Basically, the tradeoff between performance and security is implemented through resource allocation. First, the SBS has to allocate certain amounts of resources for both

IJSER

performance and security to satisfy their minimum requirements. Then, if there are more resources, the Service based systems can allocate the available resources for better performance or better security. Hence, to check whether the tradeoff is possible, we first need to make sure that the minimum performance and security requirements can be satisfied. The service based systems required that the delay should be less than traffic. The success probability of an attacker with capability *c* should be less than minimum security requirement. Service based systems only supports a limited number of security algorithms and key lengths, we can check whether both the minimum performance and security requirements can be satisfied by enumerating all supported security algorithms and key lengths to see if the above condition can be satisfied.

IV. TRADEOFF OBJECTIVE FUNCTION

When both minimum performance and security requirements are satisfied, the Service based systems can use the available resources for better performance or security. To have better security, as shown in the security metric, the Service based systems can use either a stronger algorithm with a longer key, or a larger protection percentage. Hence, to control the tradeoff between performance and security, we have to combine the performance metric and security metric together as a tradeoff objective function.

V. PERFORMANCE BIASED OBJECTIVE FUNCTION

The performance biased objective function is a tradeoff objective function that tries to maximize performance without violating the minimum security requirement. For the tradeoff objective function , the performance biased objective function sets the weighting factor of the security to 0. In this case, to minimize the tradeoff objective function is equivalent to minimizing the delay. When the encryption algorithm and the key length are fixed, the performance biased tradeoff should always use the minimum protection percentage.

V. SECURITY BIASED TRADEOFF FUNCTION

The security biased tradeoff function is a tradeoff objective function that tries to maximize security without violating the minimum performance requirements. For the tradeoff objective function, the security biased tradeoff function sets the weighting factor of the performance to 0. In this case, to minimize the tradeoff objective function is equivalent to minimizing the attacker's success probability *S*. When the encryption algorithm and the key length are fixed, we can compute the upper limit for the protection percentage from the minimum performance requirements.

VI. NON-LINEAR TRADEOFF OBJECTIVE FUNCTION

If the SBS consumer's preferences on performance and security, i.e., the weighting factors do not change with the real-time performance and security conditions, we call such a tradeoff objective function as a linear tradeoff objective function, like the performance biased tradeoff function and the security biased tradeoff function. On the contrary, if the weighting factors *a* and *b* are related to the current performance and security, we call such a tradeoff objective function as a non-linear tradeoff objective function. There may be various ways to define a non-linear tradeoff model, but all definitions should have the following properties: The weighting factor of performance increases when the performance approaches the minimum performance requirement. The minimum performance or security requirements are critical for the SBS. Hence, when the minimum performance requirements are not satisfied, the weighting factor of performance becomes infinite.

VI. CONCLUSION

So, we use an adaptive model for the tradeoff between service performance and security in service-based environments is presented. The performance and security metrics allow us to quantitatively calculate how much protection a security configuration vector can provide and how much performance will be decreased by that security configuration vector.

IJSER

REFERENCES

[1] A.C.-C. Yao, "Protocols for Secure Computations (Extended Abstract)," Proc. IEEE 23rd Symp. Foundations of Computer Science (FOCS), pp. 160-164, 1982.
[2] M. Fort, F.C. Freiling, L.D. Penso, Z. Benenson, and D. Kesdogan, "Trustedpals: Secure Multiparty Computation Implemented with Smart Cards," Proc. 11th European Symp. Research in Computer Security (ESORICS), pp. 34-48, 2006.
[3] Z. Chen, Java Card Technology for Smart Cards: Architecture and Programmer's Guide. Addison-Wesley Longman Publishing Co., Inc., 2000.
[4] N. Leavitt, "Will Proposed Standard Make Mobile Phones More Secure?," Computer, vol. 38, no. 12, pp. 20-22, Dec. 2005.
[5] Certgate GmbH, "Certgate Smart Card," http://www.certgate.
com/web_en/products/smartcardmmc.html, 2008.
[6] N.A. Lynch, Distributed Algorithms. Morgan Kaufmann Publishers, Inc., 1996.
[7] T.D. Chandra and S. Toueg, "Unreliable Failure Detectors for Reliable Distributed Systems," J. ACM, vol. 43, no. 2, pp. 225-267, 1996.
[8] F.C. Freiling, R. Guerraoui, and P. Kuznetsov, "The Failure Detector Abstraction," ACM Computing Surveys, vol. 43, no. 2, pp. 1-40, 2011.
[9] D. Malkhi and M.K. Reiter, "Unreliable Intrusion Detection in Distributed Computations," Proc. 10th Computer Security Foundations Worksop (CSFW), pp. 116-125, 1997.
[10] K.P. Kihlstrom, L.E. Moser, and P.M. Melliar-Smith, "Byzantine Fault Detectors for Solving Con sensus," Computing J., vol. 46, no. 1, pp. 16-35, 2003.ECT

IJSER