Acceleration of Secure Function Evaluation Protocol

Aram Jivanyan
American University of Armenia
Yerevan, Armenia
ajivanyan@aua.am

Gurgen Khachatryan
American University of Armenia
Yerevan, Armenia
gurgenkh@aua.am

Tigran Sokhakyan
Russian-Armenian (Slavonic) University
Yerevan, Armenia
tigran.sokhakyan@gmail.com

Davit Danoyan
Yerevan State University
Yerevan, Armenia
danoyan@gmail.com

ABSTRACT
Secure Function evaluation (SFE) is a general protocol allowing two or more mutually untruthful parties each of them holding a private input to a function $f(x_1, x_2, \dots, x_n)$ compute the function of their input without learning the real values of the input parameters. In this paper we propose an implementation of the SFE protocol based on special white-box cryptography-based oblivious transfer protocol which in several orders accelerates the OT phase and, thus, the SFE.

Keywords
Secure function evaluation, white-box cryptography, oblivious transfer, Yao's garbled circuit protocol

1. INTRODUCTION
Protocols for secure function evaluation allow two or more mutually distrustful parties to collaborate and compute some function on each other's inputs, with privacy and correctness guarantees. Andrew Yao showed that secure two-party protocols can be constructed for any computable function [1]. Yao's protocol involves representing the desired function as a Boolean circuit and having one party (often called *client* or *generator*) build the garbled circuit in such a way that it can be selectively decrypted by the other party (called *server* or *evaluator*) to compute the output. In particular oblivious transfer (OT) sub protocol is instantiated several times to obtain a subset of the decryption keys that are needed to compute output of the function.

Yao's garbled circuit technique remains one of the most promising and actively studied methods for secure multi-party computation. The very first practical implementation of secure two-party computation (2PC) [2] used Yao's basic garbled circuit approach, and it remains the primary paradigm for the plenty of 2PC implementations that have been developed during past eleven years [3] [4] [5] [6] [7].

Yao's protocol has a great practical significance. In many real-world situations, the inputs to a function may be too valuable or sensitive to share with other parties. Efficient SFE algorithms enable a variety of electronic transactions, previously impossible due to mutual mistrust of participants. Huang et al. explored the use of secure computation for biometric identification [8] in security applications, when it is desirable for individual genetic data to be kept private but still checked against a specified list. The more general case of multiparty computation has already seen real-world use in computing market clearing prices in Denmark [9]. This is not so forth full list of applications: auctions [10] [11], contract signing [12], set intersection [13].

Because the generation and execution of garbled circuit benefits from huge advances in processor speed (moreover, specialized systems have hardware blocks for cryptographic operations support) as well as the increasing availability of large numbers of cores, the computation time and cost for garbled circuit protocols has dropped dramatically. Although, many optimizations in 2PC have focused on reducing the size of the garbled circuits themselves [14] [5] [15] and reducing the number of rounds required to achieve the desired security level in case of malicious model [5], oblivious transfer (OT) protocol, which is used as a sub-protocol and many instances of it have to be executed by the larger protocol, is often too resource consuming, due to the high cost of the required public key operations. Notably, Yao's garbled-circuit protocol [1] requires OT for every input bit of one party. However, if thousands, millions or even billions of oblivious transfers need to be run, this will become prohibitively expensive.
In this paper we propose the usage of new white-box OT protocol (WB-OT) which runs several orders of magnitude faster and requires less communication bandwidth compared with the protocols currently being used [16].
Organization of the paper: In the Section 2 we cover the necessary cryptographic background, the OT protocol based on white-box cryptography. In Section 3 and in Section 4 we provide the general description of SFE protocol based on WB-OT.

2. BACKGROUND
In this section we describe the main cryptographic building blocks needed in the development of our protocol.

2.1. Yao's garbled circuits protocol

Yao's garbled circuits protocol [1] allows two parties to securely compute an arbitrary function represented in Boolean circuit. The client C garbles the gates of the Boolean circuit using symmetric keys (the length k of keys is globally fixed), which conceptually represent actual Boolean values and hide them, and sends the garbled circuit with the keys that correspond to its input bits to the server S. Later S uses a 1-out-of-2 OT (OT_2^1) to obliviously obtain the keys corresponding to its inputs and evaluates the garbled circuit gate by gate. After the evaluation S shares the output with C. Here we want to emphasize that Yao's garbled circuits protocol requires a run of OT_2^1 on strings of length k for each input bit from S.

2.2. OT protocol

The idea of OT was first introduced in 1981 by Rabin [17]. In Rabin's formulation, a sender transfers one of potentially many pieces of information to a receiver, but remains oblivious as to what piece (if any) has been transferred. In context of two-party SFE a more useful form of OT was developed later [12]. 1-out-of-2 oblivious transfer (OT_2^1) is a two-party protocol between a sender A and a receiver B. As input to OT_2^1 protocol the sender A has a pair of secret strings (s_0, s_1), and the receiver B has a selection bit i. Upon completion of protocol, B learns s_i, but nothing about $s_{i\oplus 1}$, and the sender learns nothing about i.

To face real-world applications the OT protocol extensions are developed by Ishai et al. in [18]. This approach conceptually is similar to a hybrid encryption scheme: OT extension protocol runs a small number of OTs (less than 128) and uses these runs as a base for obtaining further OTs via the use of cheap symmetric cryptographic operations only. These extensions reach relatively high performance but they do not completely get rid of expensive public-key encryption operations (anyway, the mentioned initial runs use expensive public-key encryptions). An optimized and generalized version of the former protocol was introduced in [19].

Also, we note intensive development of OT protocols which are specially optimized to be implemented on a specific hardware model taking maximum advantage of the presumed platform. Successful developments can be found in [20] [21].

2.3. White-Box Cryptography

The academic study of White-Box cryptography (WBC) was initiated in 2002 in the seminal work of Chow, Eisen, Johnson and van Oorschot [22], where a new attack context was proposed. Here an attacker has a full access to the implementation details of algorithms and the execution environment; thus the execution process, intermediate results and the algorithm internals can both be viewed and altered at will. This attack context is motivated by a variety of applications where the end-point of encryption/decryption execution is not trusted. The main goal of WBC is, therefore, to conceal the secret key used to protect data in the mentioned environment. The general technique for WBC is to build special look-up tables corresponding to the chosen key and specified functionality which is either an encryption or a decryption. During the last decade several implementations and security analysis methods of existing schemes has been proposed in academic literature so far [23] [24] [25] [26] [27]. Private sector is also interested in this field as several implementations of WBC schemes have been developed and patented by different companies including [28] [29] which highlight the real-world importance of this cryptographic discipline.

3. WB-OT PROTOCOL

One of the best known protocol extensions facing real-world application for oblivious transfer with security in the presence of semi-honest adversaries are provided by Naor and Pinkas. They present two protocols; a more efficient protocol that is secure in the random oracle model and a less efficient one that is secure in the standard model and under standard assumptions [30]. A novel WB-OT protocol introduced in [16] is a potential alternative for those desiring not to rely on random oracle model and avoid expensive public-key cryptography operations. Description of WB-OT protocol, security analysis with proofs and detailed performance analysis compared to the state-of-the-art OT protocols can be found in [16].

In our implementation of SFE protocol garbled values [1] are represented as 128 bit length strings and a white-box encryption algorithm is implemented based on SAFER+ block cipher [31].

4. SECURE FUNCTION EVALUATION BASED ON WB-OT PROTOCOL

Our implementation follows traditional development paradigms of SFE frameworks described in [2]. Some details on enhancements done can be found below.

State-of-the-art OT implementations have high cost in terms of both - network communication and processor load [32]. The evaluation of Boolean circuit having N bit length inputs requires N OT protocols to be applied. The number of public-key operations to be performed is equal to O(N) and each of them will result ciphertexts with lengths defined by the underlying cryptosystem. In modest security standards, minimum RSA-2048 should be employed leading to 2048 bit length cipher texts.

The novelty in this paper is the method of lowering the overall cost of SFE operations by reducing them in OT sub-protocol. We use the 1-out-of-2 OT protocol from [16] for decryption keys transfer corresponding evaluator's input. The applied OT protocol performs with 128 bit or 256 bit length ciphertexts and requires only white-box operations to be employed. The resulted protocol will be faster in several orders of magnitude and will require about 10 times less network bandwidth.

Except for accelerating the SFE evaluation applying white-box cryptography based OT protocol, bunch of future improvements should be done yielding to practical engine which will be applicable in large scale applications. A compiler included in our framework builds the Boolean circuit representing the function to be evaluated. As the circuit size can grow to levels that affordable hardware cannot possibly deal with, we make use of memory mapped files for storing the circuit. Further processing of circuit is done in chunks that can be stored in memory.

For further minimization of the resources consumed during garbled circuit generation and evaluation, in our next iteration we will use free-XOR [14] and four to tree garbled-row reduction [33] techniques described below.

The aim of free-XOR technique is to reduce both the network and processor load by avoiding encryption/decryption operations and transferring less data. In this method XOR gates' output evaluation is done by simply XOR-ing input wire labels. Security concerns of this technique are covered in [14].

The Garbled Row Reduction (GRR) technique deals with minimization of network load for non-XOR gates. GRR introduced by Pinkas et al. in [33] reduces the size of garbled tables for non-XOR 2-fan gates by 25% by cutting off one garbled row.

5. CONCLUSION

The framework described above is a general purpose two party secure function evaluation system secure against honest-but-curious parties. It utilizes white-box oblivious transfer protocol to achieve better performance with large input data. However, this is not secure against malicious attackers and providing security in that attack model is a milestone for our further developments.

REFERENCES

[1] A. C.-C. Yao, "How to Generate and Exchange Secrets (Extended Abstract)," in *27th Annual Symposium on Foundations of Computer Science, Toronto*, 1986.

[2] D. Malkhi, N. Nisan, B. Pinkas and Y. Sella, "Fairplay - Secure Two-Party Computation System," in *Proceedings of the 13th USENIX Security Symposium, August 9-13*, 2004.

[3] T. K. Frederiksen, T. P. Jakobsen, J. B. Nielsen, P. S. Nordholt and C. Orlandi, "MiniLEGO: Efficient Secure Two-Party Computation from General Assumptions.," in *EUROCRYPT*, 2013.

[4] Y. Huang, J. Katz and D. Evans, "Quid-Pro-Quo-tocols: Strengthening Semi-honest Protocols with Dual Execution," in *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May*, 2012.

[5] Y. Lindell and B. Pinkas, "An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries," in *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International*, 2007.

[6] Y. Lindell, B. Pinkas and N. P. Smart, "Implementing two-party computation efficiently with security against malicious adversaries," in *Security and Cryptography for Networks*, Springer, 2008, pp. 2-20.

[7] T. Schneider, Engineering Secure Two-Party Computation Protocols: Design, Optimization, and Applications of Efficient Secure Function Evaluation, Springer, 2012.

[8] D. Evans, Y. Huang, J. Katz and L. Malka, "Efficient privacy-preserving biometric identification," in *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, 2011.

[9] P. Bogetoft, D. L. Christensen, I. Damgard, M. Geisler, T. Jakobsen, M. Kroigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter and others, "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*, Springer, 2009, pp. 325-343.

[10] G. Di Crescenzo, "Private selective payment protocols," in *Financial Cryptography*, 2001.

[11] M. Fischlin, "A cost-effective pay-per-multiplication comparison method for millionaires," in *Topics in Cryptologya, CT-RSA 2001*, Springer, 2001, pp. 457-471.

[12] S. Even, O. Goldreich and A. Lempel, "A Randomized Protocol for Signing Contracts," *Commun. ACM*, vol. 28, no. 6, pp. 637-647, 1985.

[13] C. Hazay and Y. Lindell, "Constructions of truly practical secure protocols using standardsmartcards," in *Proceedings of the 15th ACM conference on Computer and communications security*, 2008.

[14] V. Kolesnikov and T. Schneider, "Improved Garbled Circuit: Free XOR Gates and Applications," in

Automata, Languages and Programming, 35th International Colloquium, 2008.

[15] V. Kolesnikov, P. Mohassel and M. Rosulek, "FleXOR: Flexible garbling for XOR gates that beats free-XOR," in *Advances in Cryptology--CRYPTO 2014*, Springer, 2014, pp. 440-457.

[16] A. Jivanyan, G. Khachatryan and A. Oliynik, "Effitient Oblivious Transfer Protocols Based on White-Box Cryptography," *PERSONAL COMMUNICATION*.

[17] M. O. Rabin, "How To Exchange Secrets with Oblivious Transfer," *Technical Report TR-81*, p. 187, 1981.

[18] Y. Ishai, J. Kilian, K. Nissim and E. Petrank, "Extending oblivious transfers efficiently," in *Advances in Cryptology-CRYPTO 2003*, Springer, 2003, pp. 145-161.

[19] V. Kolesnikov and R. Kumaresan, "Improved OT extension for transferring short secrets," in *Advances in Cryptology--CRYPTO 2013*, Springer, 2013, pp. 54-70.

[20] V. Kolesnikov, "Truly Efficient String Oblivious Transfer Using Resettable Tamper-Proof Tokens," in *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, 2010.

[21] M. Dubovitskaya, A. Scafuro and I. Visconti, "On Efficient Non-Interactive Oblivious Transfer with Tamper-Proof Hardware.," *IACR Cryptology ePrint Archive*, vol. 2010, p. 509, 2010.

[22] S. Chow, P. Eisen, H. Johnson and P. C. Van Oorschot, "White-box cryptography and an AES implementation," in *Selected Areas in Cryptography*, 2003.

[23] O. Billet, H. Gilbert and C. Ech-Chatbi, "Cryptanalysis of a white box AES implementation," in *Selected Areas in Cryptography*, 2005.

[24] Y. De Mulder, B. Wyseur and B. Preneel, "Cryptanalysis of a perturbated white-box AES implementation," in *Progress in Cryptology-INDOCRYPT 2010*, Springer, 2010, pp. 292-310.

[25] M. Karroumi, "Protecting white-box AES with dual ciphers," in *Information Security and Cryptology-ICISC 2010*, Springer, 2011, pp. 278-291.

[26] T. Lepoint and M. Rivain, "Another Nail in the Coffin of White-Box AES Implementations.," *IACR Cryptology ePrint Archive*, vol. 2013, p. 455, 2013.

[27] Y. Xiao and X. Lai, "A secure implementation of white-box AES," in *Computer Science and its Applications, 2009. CSA'09. 2nd International Conference on*, 2009.

[28] A. J. Farrugia, B. Chevallier-Mames, B. Kindarji, M. Ciet and T. Icart, "Cryptographic process execution protecting an input value against attacks". United States of America Patent 8605894 B2, 12 October 2011.

[29] P. A. Eisen, G. S. Goodes and D. E. Murdock, "System and method for generating white-box implementations of software applications". U.S. Patent CA2724793 A1, 25 May 2009.

[30] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 2001.

[31] J. L. Massey, G. H. Khachatrian and M. K. Kuregian, "Nomination of SAFER+ as candidate algorithm for the Advanced Encryption Standard (AES)," *NIST AES Proposal*, 1998.

[32] V. Kolesnikov and R. Kumaresan, "On Cut-and-Choose Oblivious Transfer and Its Variants," *Manuscript, http://people.csail.mit.edu/ranjit/papers/ccotext.pdf*, 2014.

[33] B. Pinkas, T. Schneider, N. P. Smart and S. C. Williams, "Secure Two-Party Computation Is Practical," in *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference*, 2009.