PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2016, № 1, p. 40–44

I n f o r m a t i c s

EXTENDING WHITE-BOX CRYPTOGRAPHY BASED OBLIVIOUS TRANSFER PROTOCOL

D. H. DANOYAN *

Chair of Discrete Mathematics and Theoretical Informatics YSU, Armenia

Secure computation platforms are becoming one of the most demanded cryptographic tools utilized in diverse applications, where the performance is critical. This point makes important the optimization of every component of secure computation systems. Oblivious Transfer (OT) is a fundamental cryptographic primitive heavily used in such protocols. Most of the OT protocols used today are based on public-key cryptography, hence their efficiency suffers heavily from the number of modular exponentiation operations done. OT extensions were introduced to reduce the number of basic OT protocol execution rounds requiring public-key cryptography operations. Recently a white-box cryptography based OT protocol (WBOT) was introduced that avoids using expensive public-key operations. In this article extension protocols for WBOT are presented, that further improve the novel approach by dramatically decreasing the protocol invocation count required.

MSC2010: 94A60.

***Keywords*:** secure multi-party computations, oblivious transfer extensions, white-box oblivious transfer.

Introduction. Nowadays there are billions of devices connected with each other in local networks and the internet. While a few decades ago secured data transactions were mainly used for military purposes, today the internet of things, cloud storages, online financial transactions and digital rights management problems exploit cryptographic tools and heavily depend on their security and efficiency. The diversity of applications also causes diversity of execution environments and requirements from the cryptographic tools used.

White-Box Cryptography. The purpose of white-box cryptography (WBC) is provision of security for cryptographic assets in environments, where the attacker can not only eavesdrop the communication channels, but can possibly gain advantage of using additional information such as system calls, memory

* E-mail: danoyan@gmail.com

snapshots or even algorithm and protocol full implementations. Cryptographic techniques secure in black-box context are subjects to key extraction attacks in insecure environments, so other techniques are needed. WBC, on the contrary, uses secret key for generation of the encryption/decryption tables, which are further used for the appropriate operations. Several implementations of widely used ciphers and cryptanalysis techniques [1–5] are already developed by researchers. Also such companies as Microsoft, Apple and Sony have developed and use white-box techniques and patents in their production [6–8].

Oblivious Transfer. Oblivious transfer (OT) protocol was introduced by Rabin [9]. In its initial formulation OT protocol involved two parties called S and R. In Rabin's original formulation S sends a message to R with delivery probability $1/2$, but remains oblivious whether the client received the message or not. According to another variant of OT protocol, called $1-out-of-2$ OT, introduced later by Even et al. [10], S has two messages m_0 and m_1 and R has a selection bit s. The goal of R is to receive m_s, without revealing selection bit s. The goal of S is to keep secret m_{1-s}. More generalized version of OT protocol ($1-out-of-n$ OT) was introduced in [11]. In extended version S holds n messages m_0, $m_1,\ldots,m_{n-1}$ and R has selection index $s \in [0,n)$. After execution of $1-out-of-n$ OT protocol R receives message ms and stays oblivious about all other messages, without leaking selection index s to S. Several variants of public-key cryptography based OT protocols with improvements for the ones mentioned above were developed in recent years [12, 13]. An alternative approach to these a novel white-box cryptography based OT protocol was proposed (WBOT) [14]. The protocol is designed to rely on any secure block cipher white-box implementation.

Secure Computations. Secure multi-party computation (SMC) protocol involves n parties P_0, P_2,...,P_{n-1} with respective private inputs x_1, $x_2,\ldots,x_{n-1}$ wish to compute a common agreed function f on their inputs without revealing anything, but the value $f(x_0,x_2,\ldots,x_{n-1})$. Theoretical researches on secure computation have been studied since mid 1980, after feasibility results illustrated the possibility of computation of any efficient function in securely manner. However, intensive research for bringing up secure computations to wide usage begun only a decade ago, starting from seminal implementation of Yao's garbled circuits protocol in Fairplay platform [15]. Currently there are several protocols for solving this problem. OT protocol plays crucial role in Yao's garbled circuits protocol [16], where it is used to securely exchange garbled keys between parties. $1-out-of-2$ OT protocol is executed for every input bit of one party. Another notable protocol for SMC is introduced by Goldreich, Micali and Wigderson in [17]. This protocol also works on Boolean circuits, but the evaluation of gates is done by multiple participants and communication between them heavily exploits OT.

OT Extension. OT extension protocol was introduced in [18], which extend few costly base-OTs by symmetric cryptography use only. While this protocol was of theoretical interest, Ishai et al. constructed the first known practically efficient OT extension [19]. The latter protocol confirmed that the extending Oblivious Transfer

can be done efficiently and with very little overhead. Recently, the passively secure OT extension protocol of [19] was improved by [20,21].

WBOT Extension Protocols. These protocols are considered in the random oracle model. $H : [m] \times \{0,1\}^k \to \{0,1\}^m$ and $G : \{0,1\}^k \to \{0,1\}^m$, where k is a security parameter.

Definition 1. The m-times $1-out-of-2$ WBOT functionality for l-bit vectors, denoted $m \times WBOT_l$, is defined as follows: The sender S holds m pairs of vectors $\{x_j^0, x_j^1\}$. The receiver R has m selection bits $(r_0, r_1, \ldots, r_{m-1})$. After the invocation of the protocol R should have m vectors $x_j^{r_j}$ while staying oblivious of the other vectors and R stays unaware of the selection bits.

The Protocol 0 presented below replaces $m \times WBOT_l$ with $k \times WBOT_l$, resulting the same output with less l-bit WBOT invocations.

Protocol 0:

Inputs:

S: m pairs of l-bit vectors (x_j^0, x_j^1)

R: vector of selection bits $r = (r_0, r_1, ..., r_{m-1})$

Step 1 :

S initializes a k-bit vector s randomly

R initializes a $m \times k$ matrix T with the random oracle

Step 2 :

WBOT is invoked, where R plays the sender with inputs $\{t_i, t_i \oplus r\}$, $i \in k$, and S plays the receiver with input s

Step 3 :

S sends m pairs of l-bit vectors (y_j^0, y_j^1) to R, where $y_j^0 = x_j^0 \oplus H(j, q_j)$, and $y_j^1 = x_j^1 \oplus H(j, q_j \oplus s)$, where q_j is the j^{th} row of the matrix received by S in Step 2 with columns q^i

Step 4 :

R outputs $z_j = y_j^{r_j} \oplus H(j, t_j)$.

The Protocol 1 below replaces $m \times WBOT_l$ with $k \times WBOT_k$, resulting in reduction of both, WBOT invocation count and length.

Protocol 1:

Inputs:

S: m pairs of l-bit vectors (x_j^0, x_j^1)

R: vector of selection bits $r = (r_0, r_1, \ldots, r_{m-1})$

Step 1 :

S initializes a k-bit vector s randomly

R initializes k pairs of random k-bit vectors $\{k_i^0, k_i^1\}$

WBOT is invoked, where R plays the sender with inputs $\{k_i^0, k_i^1\}$ $i \in [0,k)$ and S plays the receiver with input s

Step 2 :
R constructs $m \times k$ matrix T, where $t^i = G(k_i^0)$
R sends $u^i = t^i \oplus G(k_i^1) \oplus r$ to S for all $i \in [0,k)$
S constructs $m \times k$ matrix Q, where $q^i = s_i * u^i \oplus G(k_i^{s_i})$
Step 3 :
S sends m pairs of l-bit vectors (y_j^0, y_j^1) to R, where $y_j^0 = x_j^0 \oplus H(j, q_j)$, and $y_j^1 = x_j^1 \oplus H(j, q_j \oplus s)$, where q_j is the j^{th} row of the matrix received by S in Step 2 with columns q^i
Step 4 :
R outputs $z_j = y_j^{r_j} \oplus H(j, t_j)$.

Provided with WBOT proved secure with semi-honest participants (both parties follow the protocol, but try get more information "legally") [14] it is easy to show that these extensions do not reduce the security in the random oracle model. Below it is provided the proof of correctness of Protocol 1. Protocol 0 can be proved correctly with similar considerations.

Proof of Correctness. What we need to prove is that $z_j = x_j^{r_j}$.
First of all lets find a relation between q_j and t_j:
in Step 2 $q^i = s_i * u^i \oplus G(k_i^{s_i})$, where $u^i = t^i \oplus G(k_i^1) \oplus r$ and $t^i = G(k_i^0)$,
from these three formulas we get
$q^i = s_i * (G(k_i^0) \oplus G(k_i^1) \oplus r) \oplus G(k_i^{s_i}) = s_i * G(k_i^0) \oplus s_i * G(k_i^1) \oplus s_i * r \oplus G(k_i^{s_i})$.
When $s_i = 0$, $\quad s_i * G(k_i^0) \oplus s_i * G(k_i^1) \oplus G(k_i^{s_i}) = G(k_i^0)$
and $s_i = 1$, $\quad s_i * G(k_i^0) \oplus s_i * G(k_i^1) \oplus G(k_i^{s_i}) = G(k_i^0)$,
so $q^i = s_i * r \oplus G(k_i^0) = s_i * r \oplus t^i$, therefore, $q_j = s * r_j \oplus t_j$.
Now lets apply this result to outputs $z_j = y_j^{r_j} \oplus H(j, t_j)$:
Case $1 : r_j = 0$
$z_j = y_j^0 \oplus H(j, t_j)$ and from Step 3 $y_j^0 = x_j^0 \oplus H(j, q_j)$, so
$z_j = x_j^0 \oplus H(j, q_j) \oplus H(j, t_j) = x_j^0 \oplus H(j, s * r_j \oplus t_j) \oplus H(j, t_j) = x_j^0$.
Case $2 : r_j = 1$
$z_j = y_j^1 \oplus H(j, t_j)$ and from Step 3 $y_j^1 = x_j^1 \oplus H(j, q_j \oplus s)$, so
$z_j = x_j^1 \oplus H(j, q_j \oplus s) \oplus H(j, t_j) = x_j^1 \oplus H(j, s * r_j \oplus t_j \oplus s) \oplus H(j, t_j) = x_j^1$.
In both $z_j = x_j^{r_j}$, so Protocol 1 works correctly.

Conclusion. In this article two extensions are provided for white-box cryptography based OT protocols. These protocols are mainly applied to speed up secure multi-party computations. The number of actual protocol invocations required is decreased in order of magnitude depending on the security parameter provided. These extensions prove the possibility of construction of secure OT protocols, which are not based on public key cryptography primitives. It is the next huge milestone to change the protocols for widening adversary model, where the provided protocol extensions are secure without loss of their efficiency.

Received 15.02.2015

REFERENCES

1. **Chow S., Eisen P., Johnson H., van Oorschot P.C.** White-Box Cryptography and an Aes Implementation. // Selected Areas in Cryptography, 2003, p. 250–270.
2. **Chow S., Eisen P., Johnson H., van Oorschot P.C.** A White-Box Des Implementation for Drm Applications. Digital Rights Management, 2003, p. 1–15.
3. **Link H.E., Neumann W.D.** Clarifying Obfuscation: Improving the Security of White-Box DES. Information Technology. "Coding and Computing, 2005". 2005, p. 679–684.
4. **Billet O., Gilbert H., Ech-Chatbi C.** Cryptanalysis of a White Box Aes Implementation. // Selected Areas in Cryptography, 2004, p. 227–240.
5. **Wyseur B., Michiels W., Gorissen P., Preneel B.** Cryptanalysis of White-Box Des Implementations with Arbitrary External Encodings. // Selected Areas in Cryptography, 2007, p. 264–277.
6. **Eisen P., Goodes G., Murdock D.E.** System and Method for Generating White-Box Implementations of Software Applications. US Patent Application CA2724793 A1, 2009.
7. **Michiels W., Gorissen P.** Cryptographic Method for a White-Box Implementation. US Patent Application WO2008059420 A2, 2007.
8. **Farrugia A.J., Chevallier-Mames B., Kindarji B., Ciet M., Icart T.** Cryptographic Process Execution Protecting an Input Value Against Attacks. US Patent Application US 8605894 B2, 2011.
9. **Rabin M.** How to Exchange Secrets by Oblivious Transfer. Harvard U., Tech. Memo TR-81, Aiken Computation Laboratory, 1981.
10. **Even S., Goldreich O., Lempel A.** A Randomized Protocol for Signing Contracts. // Communications of the ACM, 1985, v. 28, №6, p. 637–647.
11. **Brassard G., Crepeau C., Robert J.-M.** All-Or-Nothing Disclosure of Secrets. // Advances in Cryptology–CRYPTO'86, 1986, p. 234–238.
12. **Lindell Y., Pinkas B.** Secure Two-Party Computation Via Cut-and-Choose Oblivious Transfer. // Journal of Cryptology, 2012, v. 25, №4, p. 680–722.
13. **Naor M., Pinkas B.** Efficient Trace and Revoke Schemes. // Financial Cryptography, 2001, p. 1–20.
14. **Jivanyan A., Khachatryan G., Oliynik A.** Efficient Oblivious Transfer Protocols Based on White-Box Cryptography. AUA Internal Reports, 2013.
15. **Malkhi D., Nisan N., Pinkas B., Sella Y.** Fairplay – A Secure Two-Party Computation System. USENIX Security Symposium, 2004, v. 4 .
16. **Yao A.** How to Generate and Exchange Secrets. Foundations of Computer Science, 27th Annual Symposium, 1986, p. 162–167.
17. **Goldreich O., Micali S., Wigderson A.** How to Play any Mental Game. Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, 1987, p. 218–229.
18. **Beaver D.** Correlated Pseudo-Randomness and the Complexity of Private Computations. Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, p. 479–488.
19. **Ishai Y., Kilian J., Nissim K., Petrank E.** Extending Oblivious Transfers Efficiently. // Advances in Cryptology–CRYPTO'2003, 2003, p. 145–161.
20. **Asharov G., Lindell Y., Schneider T., Zohner M.** More Efficient Oblivious Transfer and Extensions for Faster Secure Computation. Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, 2013, p. 535–548.
21. **Kolesnikov V., Kumaresan R.** Improved OT Extension for Transferring Short Secrets. // Advances in Cryptology–CRYPTO'2013, 2013, p. 54–70.