Efficient Oblivious Transfer Protocols based on White-Box Cryptography

Aram Jivanyan[1], Gurgen Khachatrian, Andriy Oliynyk[2], and Mykola Raievskyi

[1] American University of Armenia
40 Marshal Baghramyan Ave. Yerevan 0019, Republic of Armenia
{ajivanyan,gurgenkh}@aua.am
http://www.aua.am
[2] Samsung Ukraine R&D Institute, Samsung Electronics Ukraine Co.,LLC
57, Lva Tolstogo St., Kyiv 01032, Ukraine
{a.oliynyk,m.raievskyi}@samsung.com

Abstract. Oblivious transfer protocol is an important cryptographic primitive having numerous applications and particularly playing an essential role in secure multiparty computation protocols. On the other hand existing oblivious transfer protocols are based on computationally expensive public-key operations which remains the main obstacle for employing such protocols in practical applications. In this paper a novel approach for designing oblivious transfer protocols is introduced based on the idea of replacing public-key operations by white-box cryptography techniques. As a result oblivious transfer protocols based on white-box cryptography run several times faster and require less communication bandwidth compared with the existing protocols.

Keywords: oblivious transfer, white-box cryptography, secure function evaluation

1 Introduction

A secure function evaluation(SFE) protocol for a computable function $f(x, y)$ enables two parties, Alice who owns x and Bob, who owns y, to compute the value $f(x, y)$ in a way, that does not reveal to each side more information that can be deduced from the computed result $f(x, y)$ [1]. One of the main results in cryptographic research shows that for each polynomially computable function $f(\cdot, \cdot)$ there exists such a (polynomially computable) protocol ([2]). The main building block used for implementing such functionality is the so-called Oblivious Transfer(OT) protocol. Despite the fact that the secure function evaluation protocol has polynomially complexity, the resulting protocols often are not as efficient since the number of required OT protocols to be executed is proportional to the size of the circuit computing the given function $f(\cdot, \cdot)$. Even for relatively simple functions this can be prohibitively expensive as each OT instance requires several public-key operations to be performed. More specifically 1-out-2 OT protocol is the following: Alice (Server) has in possession two data

items k_1 and k_2. Bob (Receiver) chooses secretly which item to obtain. OT is a protocol how Bob and Alice interact in a way that as a result of that interaction Bob gets the item he chooses and has no information about the other data item; meanwhile, Alice has no idea which of data items k_1 or k_2 Bob has received. In this paper we explore a novel technique for implementing efficient OT protocols which operate significantly faster and also require less communication bandwidth. The idea is based on using white-box(WB) cryptography as an alternative to ordinary public-key cryptography schemes for securing two-party communications. White-box cryptography allows to implement symmetric cryptographic functionality so it can be executed in the untrusted domains and the same time not violate the secret key privacy. The general technique is to built special look-up tables corresponding to the secret key which will allow to make encryption without revealing the key itself. White-box cryptographic schemes are mainly used in DRM systems[4] and software protections methods[5]. The secure white-box implementation of encryption functionality allows the secret key owner to share the white-box implementation with third parties so they will be able to encrypt any message with the specified secret key without learning the key itself. The encrypted message can be decrypted only by the secret key owner. These considerations allow us to employ these techniques for enabling secure communication between the participating parties of secure function evaluation protocol. As such the contribution of this paper is twofold. Firstly a new method for constructing more efficient OT protocols is introduced. Secondly an idea to use white-box cryptography in the new context, namely to replace public-key operations by white-box operations is introduced. To our best knowledge this has been accomplished for the first time.

Organization of the paper: In the next section OT protocols as well as white-box cryptography techniques are discussed in more details. Section 3 presents new designs of two OT protocols based on white-box cryptography. The security of the proposed protocols are discussed in Section 4 and some experimental results about their efficiency are given in Section 5.

2 Preliminaries

In this section the OT protocols and white-box cryptography techniques are covered in more details. Security definitions for OT protocol on which the security proofs of our designs are based, are also presented.

2.1 Oblivious Transfer

OT was first introduced by Rabin [6]. In Rabins formulation, the Server sends a message to Client with probability 1/2 but remains oblivious as to whether the Client received the message. A closely related variant called 1-out-of-2 OT was later introduced and discussed by Even, Goldreich and Lempel[7]. In their setting, Alice (the Server) has two bits b_0 and b_1, and Bob (the Client) has

a selection bit s. The goal is for Bob to receive b_s, and remain oblivious of b_{1-s} while Alice remains oblivious of s. Nowadays OT protocol stands for the notion introduced by Even et al. Essentially every known suggestion of public-key cryptography can be used to construct OT protocol and the complexity of 1-out-of-2 OT protocol is comparable to public-key operations. Brassard, Crepeau and Santha [8] extended the basic notion of 1-out-of-2 OT to 1-out-of-N OT. Namely the Server has N messages, and the Client is allowed to learn exactly one of them, while the Server is required to remain oblivious regarding the Clients selection. They gave a method for constructing 1-out-of-N OT protocols from $N-1$ invocations of a 1-out-of-2 OT protocol. More efficient implementations for 1-out-of-N OT protocol were later proposed by Naor and Pinkas [11] which require only $logN$ invocations of 1-out-of-2 OT . So the input and output of 1-out-of-N OT protocols are defined as follows.

- INPUT
  - Client: An index $0 \leq \sigma \leq N-1$
  - Server: N data elements $k_0, k_1, \ldots, k_{N-1}$
- OUTPUT
  - Client: k_σ
  - Server: Nothing

OT is a two party protocol and the most efficient implementations usually require two round communication between the participating parties. The high level overview of 1-out-of-N OT protocol is described in Fig. **??**.

The first step called **Query Preparation** is executed by Server and the step's outputs are sent to Client as an auxiliary information for generating query. The second phase is the **Query Generation** phase carried by Client where he/she generates its query and sends it to the Server. Getting the query, the Server responses to it by generating **Query Response**. Then the Client reveals its query from the sent response with help of **Response Processing** method. Almost all basic OT protocol implementations share these steps although in some cases the output of **Query Preparation** can be empty.

OT protocol has been generalized in many different ways.The next step in extending the notion of OT was k-out-of-N OT. In such protocols, the Server holds a set of N messages; she is willing to allow the Client to learn any k messages of the set, but she refuses to allow the Client to learn any information regarding the remaining N-k messages. The Client on the other hand demands that Alice remains oblivious regarding his selection of k messages. Constructions for k-out-of-N OT were presented in [9]. Another version of k-out-of-N OT is OT with Adaptive Queries firstly introduced by Naor and Pinkas [10]. In this context, the Server has N values, and the Client would like to learn k of them, deciding which ones in an adaptive manner, i.e. the i-th value may depend on the first i-1 received values. The advantage of the adaptive scheme is that the number of

Clients queries need not be prefixed or known before the execution of the protocol. Naor and Pinkas have presented efficient protocols for this problem that require O(N) computation in the preprocessing stage and fixed computation for each new value the Client obtains.

In this paper efficient solutions for 1-out-of-N OT protocol ($N \geq 2$) will be introduced.

Correctness and Security Definitions. The definition of OT protocol correctness is simple: At the end of successful execution where both Client and Server follows the protocol, the Client should obtain the value X_σ of his choice σ. Next we follow to the [11] for giving the security definitions. While defining security for OT protocol we will threat 1-out-of-2 OT protocol as 1-out-of-N OT protocol where $N = 2$. The security of OT protocol is considered usually under the honest-but-curious attacking model, where all parties honestly follows to the protocol, but they try to extract the other partys secrets. To define security of OT protocol, we separately discuss the security of Server and the security of Client.

The Client Security: For any $\sigma, \tau \in \{0, 1, \ldots, N-1\}$ and for any probabilistic polynomial time adversary B executing the Servers part, the information that B sees in case the Client tries to obtain k_σ and the information that B sees in case the Client tries to obtain k_τ are computationally indistinguishable given $k_0, k_1, \ldots, k_{N-1}$.

The Server Security: The Client must not learn or get more or different information than he should. In the IDEAL model it is assumed that there is a trusted third party T which receives $k_0, k_1, \ldots, k_{N-1}$ from the Server and receives the choice bit σ from the Client and tells the Client the value k_σ. For every distribution on the inputs $k_0, k_1, \ldots, k_{N-1}$ and any adversarial probabilistic polynomial-time machine A that plays the role of Client in the REAL model, there exists a simulator $A\prime$ which plays the Clients role in the IDEAL model and receives the same information about $k_0, k_1, \ldots, k_{N-1}$ as A, such that the outputs of A and $A\prime$ are computationally indistinguishable.

2.2 White-Box Cryptography

White-box cryptography concerns the design and analysis of implementations of cryptographic algorithms engineered to execute on untrusted platforms. Such implementations are said to operate in a an attack model where all details of the implementation are completely visible to an attacker: not only do they see input and output, they see every intermediate computation that happens along the way. This is called white-box attack context. The goal of a white-box attacker when targeting an implementation of a cipher is typically to extract the cryptographic key;thus, white-box implementations have been designed to thwart this goal (i.e., to make key extraction diffcult/infeasible). The academic

study of white-box cryptography was initiated in 2002 in the seminal work of Chow, Eisen, Johnson and van Oorschot [15].In their seminal work, they motivated and defined the white-box attack context and presented some generic techniques that can be used to help create cryptographic implementations that resist key-extraction[15] [16]. The general technique is to built special look-up tables corresponding to the chosen key and specified functionality which is either encryption or decryption. Dozens of novel implementations and security analysis of existing schemes has been proposed in academic literature so far [17][18][19] [22] [20] [21]. Yet more implementations of white-box encryption schemes has been developed and patented by different private companies including Apple, SAMSUNG, Irdeto [23][24][25] which shows the high importance of this cryptographic primitive.
Let us fix the key space K, message space M and ciphertext space C and assume that secure symmetric encryption algorithm is a pair of algorithms *(Enc, Dec)* such that $Enc : K \times M \to C$ and $Dec : K \times C \to M$. In general $K = \{0,1\}^{128}$ or $K = \{0,1\}^{256}$ and $M = C = \{0,1\}^{128}$. We consider white-box encryption scheme as a set of three algorithms **(Gen, Enc, Dec)** performing in the key space K, message space M and ciphertext space C such that given a master secret key $s \leftarrow_R K$, the algorithms performs as follows:

- **Gen:** The owner of the secret key s generates white-box encryption tables corresponding to the secret key s using the white-box table generation algorithm **Gen**.
- **Enc:** A plaintext $P \leftarrow_R M$ randomly chosen from the message space can be encrypted given the white-box tables T and the white-box encryption algorithm **Enc**: Note that $\mathbf{Enc}(T, P) = Enc(s, P)$. We will also use the notation $\mathbf{Enc}_T(P)$ for $\mathbf{Enc}(T, P)$.
- **Dec:** A ciphertext chosen from the ciphertext space C can be decrypted using the secret key s and the black-box decryption algorithm **Dec**: Actually $\mathbf{Dec} \approx Dec$.

There is no formal framework for proving any white-box encryption scheme security. Intuitively white-box scheme can be considered secure if no computationally bounded adversary will be able to extract the master encryption key from the white-box encryption tables generated with help of algorithm **Gen**. Neither any adversary should be able to make decryption functionality with help of only the white-box encryption tables.
As such white-box scheme is considered to be secure if it is secure against these key-recovery and reverse-engineering attacks.

3 Design of New OT Protocols based on White-Box Cryptography

In this section two different OT protocols based on white-box cryptography(WB-OT) are presented which designs are inspired from the most efficient and well known OT protocols. The first WB-OT protocol shares basic principles with the

RSA-based OT protocol[7] which will be referred as EGL-OT. The second one is inspired from the Naor and Pinkas OT protocol [11] which is based on El-Gamal cryptosystem and later will be referred as NP-OT.
While giving the new protocol designs, it is assumed that any secure white-box scheme based on AES, SAFER+, SERPENT, IDEA, BLOWFISH or other secure block cipher can be considered as a candidate to be employed in our designs. Hereafter we will assume the existence of secure white-box encryption scheme **(Gen, Enc, Dec)** without specifying the underlying symmetric block-cipher encryption algorithm.

3.1 1-out-of-N OT Protocol based on EGL-OT

The first protocol presented here is a general 1-out-of-N OT protocol inspired by the work [7] where N$\geq$ 2.

- ◇ **Input:** The Clients input is a choice index $\sigma \in 0, 1, , N-1$ and the Servers input is N l-bit secrets k_0, k_1, k_{N-1} where $1 \leq l \leq 128$ and l is known also to Client.

- ◇ **Auxiliary Input:**: The Server generates a master block-cipher key S for the chosen block cipher algorithm and then uses white-box table generation function **Gen** to generate corresponding white-box encryption tables $T = \mathbf{Gen}(S)$ related to the secret key S. Note that having these tables, the Client will be able to compute encryption functionality without owning the secret key S. The white-box encryption tables T are sent to the Client. S is kept secret by Server.

- ◇ **Protocol:**The protocol steps are the following.

  - **Query Preparation Phase:**The Server generates N random values $m_0, m_1, \ldots, m_{N-1} \in \{0,1\}^{128}$ and sends them to the Client.
  - **Query Generation Phase:** The Client generates a random value $r \in \{0,1\}^{128}$ and then calculates its encryption $\mathbf{Enc}_T(r)$ with help of the white-box encryption tables T. Next the computed value is used to blind the value m_σ in the following way $V = m_\sigma \bigoplus \mathbf{Enc}_T(r)$: , where $\bigoplus$ stands for XOR operation of two 128-bit vectors. V is sent to the Server.
  - **Query Response Phase:** The Server takes V to compute N values $r_0, r_1, \ldots, r_{N-1}$ related to values $m_0, m_1, , m_{N-1}$ in the following way: $r_0 = \mathbf{Dec}_S(V \bigoplus m_0), r_1 = \mathbf{Dec}_S(V \bigoplus m_1) \ldots, r_{N-1} = \mathbf{Dec}_S(V \bigoplus m_{N-1})$. Next $k'_0 = \bar{\bar{k}}_0 \bigoplus r_0, k'_1 = \bar{\bar{k}}_1 \bigoplus r_1, ..., k'_{N-1} = \bar{\bar{k}}_{N-1} \bigoplus r_{N-1}$ values are calculated and sent to Client, where $\bar{\bar{k}}_i = k_i \| \overbrace{00.000}^{128\text{-}l}$. The decryption operation **Dec** is performed with the help of the the secret key S.
  - **Response Processing Phase:** The Client computes $\bar{\bar{k}}_s = k'_\sigma \bigoplus r$ and takes its l left bits as desired secret k_s.

The general method for constructing a 1-out-of-N OT protocol having the 1-out-of-2 OT instance shown in [11] requires logN invocations of 1-out-of-2 OT protocol and NlogN calls of pseudorandom functions which are modeled by block cipher algorithms. Since white-box operations are comparable with the block cipher operations from the efficiency point of view, the presented design of 1-out-of-N protocol can operate faster for $N > 2$ as it requires only N block cipher operations and one white-box operation to be performed.

3.2 1-out-of-2 OT Protocol based on NP-OT

The next is a new 1-out-of-2 OT protocol design which was inspired by the work [11].

- ◇ **Input:** The Clients input is a choice index $\sigma \in 0, 1$ and the Servers input is 2 l-bit secrets k_0 and k_1 where $1 \leq l \leq 128$ and l is known also to Client.

- ◇ **Auxiliary Input:**: The Server generates a master block-cipher key S for the chosen block cipher algorithm and then uses white-box table generation algorithm **GEN** to generate corresponding white-box encryption tables $T = \mathbf{Gen}\ (S)$ related to the secret key S. Note that having these tables, the Client will be able to compute encryption functionality without owning the secret key S. The white-box encryption tables T are sent to the Client. S is kept secret by Server.

- ◇ **Protocol:**The protocol steps are the following.

  - **Query Preparation Phase:**The Server generates a random value $c \in \{0,1\}^{128}$ and sends it to the Client.
  - **Query Generation Phase:** The Client generates one random value $r \in \{0,1\}^{128}$ and then calculates two values $P_\sigma = \mathbf{Enc}_T(r)$ and $P_{1-\sigma} = c \bigoplus \mathbf{Enc}_T(r)$ with help of the white-box encryption tables T. Next he sends P_0 to Server.
  - **Query Response Phase:** The Server takes P_0 , derives $P_1 = C \bigoplus P_0$ and then calculates $E_0 = \bar{\bar{k}}_0 \bigoplus \mathbf{Dec}_S(P_0)$ and $E_1 = \bar{\bar{k}}_1 \bigoplus \mathbf{Dec}_S(P_1)$. Here $\bar{\bar{k}}_0 = k_0 \| \overbrace{00.000}^{128\text{-}l}$ and $\bar{\bar{k}}_1 = k_1 \| \overbrace{00.000}^{128\text{-}l}$. The decryption operation **Dec** is performed with help of the the secret key S. The values E_0 and E_1 are sent to Client.
  - **Response Processing Phase:** The Client computes $\bar{\bar{k}}_\sigma = E_\sigma \bigoplus r$ and takes its l left bits as the desired secret k_σ.

This protocol can be extended to the general 1-out-of-N instance by using the technique described in [11].

4 Correctness and Security Analysis

The correctness proofs for the specified protocols are straightforward.

Recalling the security definitions given in Section 1.1 we should separately discuss the Client's and Server's security aspects for both WB-OT protocols described above.

The Clients Security: For any $\sigma, \tau \in \{0, 1, \ldots, N-1\}$ and for any probabilistic polynomial time adversary B executing the Servers part, the information that B sees in case the Client tries to obtain k_σ and the information that B sees in case the Client tries to obtain k_τ are computationally indistinguishable given $k_0, k_1, \ldots, k_{N-1}$.

Let us start from the Client's security in the 1-out-of-N OT Protocol based on EGL-OT. In our honest-but-curious adversarial model the Server gains the value $V = m_\sigma \bigoplus \mathbf{Enc}_T(r)$ in the case the Client wants to obtain k_σ and $V\prime = m_\tau \bigoplus \mathbf{Enc}_T(r)$ in the case the Client wants to obtain k_τ. If the value r has uniform distribution, so has the value $\mathbf{Enc}_T(r)$. This fact comes from the computational security property of underlying block cipher encryption algorithm, which means that for every message x the distribution $Enc_S(U_M)$ is pseudorandom. Here U_M means a uniformly random selection of message from the message space. It can be concluded that the values V and V' both have uniform distribution so they are computationally indistinguishable. B should differentiate also the values $r_0 = \mathbf{Dec}_S(V \bigoplus m_0), r_1 = \mathbf{Dec}_S(V \bigoplus m_1) \ldots, r_{N-1} = \mathbf{Dec}_S(V \bigoplus m_{N-1})$ one of which is equal to r and the others are equal to $\mathbf{Dec}_S(\mathbf{Enc}_T(r) \bigoplus m_\sigma \bigoplus m_j)$. From the security properties of the underlying block cipher algorithm which means that for every ciphertext y the distribution $Dec_{U_K}(y)$ is pseudorandom, the uniformly random value r is computationally indistinguishable from the value $\mathbf{Dec}_S(\mathbf{Enc}_T(r) \bigoplus m_\sigma \bigoplus m_j)$. This means that the information B gets in the case when the Client tries to obtain k_σ and the information that B gets in the case when the Client tries to obtain k_τ are computationally indistinguishable given $k_0, k_1, \ldots, k_{N-1}$.
The Client's security in the 1-out-of-2 OT Protocol based on NP-OT can be proven by similar considerations.

The Servers Security: The Client must not learn or get more or different information than he should. In the IDEAL model let us assume that there is a trusted third party T that receives $k_0, k_1, \ldots, k_{N-1}$ from the Server and receives the choice bit σ from the Client and tells the Client the value k_σ. For every distribution on the inputs $k_0, k_1, \ldots, k_{N-1}$ and any adversarial probabilistic polynomial-time machine A that plays the role of Client in REAL model, there exists a simulator $A\prime$ which plays the Clients role in the IDEAL model and receives the same information about $k_0, k_1, \ldots, k_{N-1}$ as A, such that the outputs of A and $A\prime$ are computationally indistinguishable.

Without loss of generality let us assume that the data items have 128-bit length, they have uniform distribution, and the Clients choice is $\sigma = 0$. In the

IDEAL model the $A\prime$ learns only the secret value k_0 and can only guess the values of k_i where $0 < i \leq N-1$. In the REAL model the A learns the following values:

- ◇ The white-box encryption tables T.
- ◇ N random values $m_0, m_1, \ldots, m_{N-1} \in \{0,1\}^{128}$
- ◇ N values k_0 and

$$k'_1 = \bar{\bar{k}}_1 \bigoplus \mathbf{Dec}_S(\mathbf{Enc}_T(r) \bigoplus m_0 \bigoplus m_1),$$
$$\vdots$$
$$k'_{N-1} = \bar{\bar{k}}_{N-1} \bigoplus \mathbf{Dec}_S(\mathbf{Enc}_T(r) \bigoplus m_0 \bigoplus m_{N-1})$$

The security properties of the underlying block cipher algorithm implies that $Dec_{U_K}(y)$ has pseudorandom distribution so each of the received values $k'_i = \bar{\bar{k}}_i \bigoplus \mathbf{Dec}_S(\mathbf{Enc}_T(r) \bigoplus m_0 \bigoplus m_i)$ can not be distinguished from a pseudo-randomly chosen value by any observer who does not own the decryption key s. This implies that the values $k'_1, ..., k'_{N-1}$ does not reveal any information about corresponding data items $k_1, ..., k_{N-1}$ unless the A is not able to decrypt the values $\mathbf{Enc}_T(r) \bigoplus m_0 \bigoplus m_i$ for $i = 1, ..., N-1$. As the underlying white-box cryptosystem is secure against key-extraction and reverse-engineering attacks, the Client can not extract the secret key s from the white-box tables T neither can use the white-box encryption tables for decrypting the given messages $\mathbf{Enc}_T(r) \bigoplus m_0 \bigoplus m_i$. Thereby the adversary A does not learn any information about the data items $k_1, ..., k_{N-1}$ which has not been queried and thus the Server's security is preserved.
The Server's security in the 1-out-of-2 OT Protocol based on NP-OT can be proven by similar considerations.

5 Performance Analysis and Applications

For performing comparison between our proposed OT protocols and other public-key based protocols we have implemented a white-box encryption algorithm based on SAFER+ block cipher[26]. In our implementation the encryption tables have about 1MB size and the white-box encryption operations are about 4 times slower compared with black-box encryption. The experiments had been carried on Intel(R) Core 2 CPU 430 1.80 GHz * 1.80 GHz computer. The experiments are carried for the 1-out-of-2 OT instances and in both presented 1-out-of-2 OT protocols the Server needs to perform two symmetric decryption operations and the Client computes one white-box encryption operation. For comparison we have used the OpenSSL RSA implementation with 1024 bit length keys. Let us assume that the Server's data items are 16 byte length blocks which is a common case in secure function evaluation protocols where the data items represents block cipher encryption keys. In our experiments the white-box encryption for one block data (16-byte) takes about 0.00488ms and the RSA encryption of 16 byte block takes about 0.2324ms. The black-box decryption for one block data takes 0.00122ms and the RSA decryption for one block data takes 4.4ms. As

such in white-box based OT the overall speed of these computations will be 0.0049+2*0.0012=0.0073ms, and in RSA based OT will be 0.2324 + 2 * 4.4 = 9.03ms. This means the white-box based OT is about 1200 times faster then RSA based OT. Now let us see, that in this setting where 1024-bit RSA is used and the data items are 128-bit length, we get also about 8 times less communication bandwidth compared with the RSA-based OT protocol. The data items can be longer, but block encryption will not expand the message unlike public-key encryption. It should be mentioned that an acceleration of the OT operations by thousand times is not a limit since faster white-box schemes can be developed. Another argument is that the key size of public key schemes are increasingly becoming larger which is not the case for symmetric key cryptography.

6 Conclusion

White-box cryptography is a relatively new cryptographic primitive being found to be a critical component for designing secure DRM and software protection systems. In this paper it was shown how white-box cryptography can be used to significantly speed-up OT protocol. Our approach allows to accelerate OT protocol several orders of magnitude and thus make secure function evaluation protocols practical for real-life applications. It was shown how the security of the protocols proposed in this paper depends only on the security of underlying white-box encryption scheme. Thus assuming the existence of secure white-box encryption scheme a much more computationally efficient and secure OT protocols can be implemented.
One of the possible directions for future research would be the design of different generalizations of OT protocols based on white-box cryptography. It is also an interesting work to investigate the security of the proposed protocols in the case of malicious adversarial model [14].

Acknowledgments. The research was supported by Samsung Electronics.

References

1. A. Yao. How to Generate and Exchange Secrets. In 27th FOCS, pages 162-167, 1986.
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, 1987, pp.218-229.
3. J. Kilian.: Founding Cryptography on Oblivious Transfer. In 20th STOC, pages 20-31 (1988)
4. W. Michiels and P. Gorissen. "Cryptographic Method for a White-Box Implementation". U.S. Patent Application 2010/0080395 A1, Filed November 9, 2007.
5. W. Michiels and P. Gorissen. "Cryptographic System". U.S. Patent Application 2011/0116625 A1, filed March 2, 2009.
6. M. Rabin. How to Exchange Secrets by Oblivious Transfer. Tech. Memo TR-81, Aiken Computation Laboratory, Harvard U., 1981.

7. S. Even, O. Goldreich and A. Lempel. A randomized protocol for signing contracts: Communications of the ACM, Vol. 28, 1985, pp. 637647.
8. G. Brassard, C. Crepeau and J.M. Robert. All-or-nothing disclosure of Secrets: Advances in Cryptology - Crypto 86, Lecture Notes in Computer Science (LNCS) 263, Springer Verlag, 1987, pp. 234238
9. M. Naor and B. Pinkas. Computationally secure oblivious transfer. Journal of Cryptology, 18, 2005, pp. 135.
10. M. Naor, B. Pinkas, Oblivious Transfer with Adaptive Queries In Proceedings of Advances in Cryptology CRYPTO 99 . Lecture Notes in Computer Science, vol. 1666, Springer-Verlag, New York. 573 590
11. M. Naor, B. Pinkas. Effcient Oblivious Transfer protocols. In SODA 01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, pages 448457, Philadelphia, PA, USA, 2001.
12. Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications, Proc. of ISTCS97, IEEE Computer Society, 1997, pp. 174-184.
13. T. Tassa. Generalized OTby Secret Sharing. Designs, Codes and Cryptography , 58(1), pp. 11-21 2011)
14. Y. Lindell. Efficient Fully-Simulatable OT Protocol In CT-RSA, 2008.
15. S. Chow, P. Eisen, H. Johnson, and P.C. van Oorschot: White-Box Cryp- tography and an AES Implementation. In Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002, LNCS 2595(2003), pages 250-270.
16. S. Chow, P. Eisen, H. Johnson, and P.C. van Oorschot: A White-box DES Implementation for DRM Applications. In Digital Rights Management: ACM CCS-9 Workshop, DRM 2002", LNCS 2696(2003), pages 1-15.
17. O. Billet, H. Gilbert, C. Ech-Chatbi. Cryptanalysis of a White-box AES Implementation. In Selected Areas in Cryptography 2004 (SAC 2004), pages 227-240
18. T.Lepoint, M. Rivain Another Nail in the coffin of White box AES implementations 2013. http://eprint.iacr.org/2013/455
19. M. Karroumi. Protecting white-box AES with Dual Cipher. ICISC 2010, volume 6829 of Lecture Notes in Computer Science, pages 278291. http://dx.doi.org/10.1007/978-3-642-24209-0_19
20. Y. De Mulder, P. Roelse, B. Preneel. Cryptanalysis of a perturbated white-box AES implementation. INDOCRYPT 2010,volume 6498 of Lecture Notes in Computer Science, pages 292310. http://dx.doi.org/10.1007/978-3-642-17401-8_21
21. Y. Xiao,X. Lai. A secure implementation of white-box AES. In Computer Science and its Applications, 2009. CSA09. 2ndInternational Conference on, pages 16. IEEE, 2009.
22. Y. De Mulder, P. Roelse, B. Preneel. Cryptanalysis of the Xiao Lai white-box AES implementation. Selected Areas in Cryptography, volume7707 of Lecture Notes in Computer Science, pages 3449, 2013. URL: http://dx.doi.org/10.1007/978-3-642-35999-6_3.
23. P. Eisen, G. Goodes, D. E. Murdock.System and method for generating white-box implementations of software applications, U.S. Patent Application CA2724793 A1, filed May 25, 2009.
24. W. Michiels, P. Gorissen. Cryptographic method for a white-box implementation,U.S. Patent Application WO2008059420 A2, filed Nov 9, 2007.
25. Cryptographic process execution protecting an input value against attacks U.S. Patent Application US 8605894 B2, filed Oct 12, 2011.
26. J.Massey, G.Khachatrian, M.Kuregian. Nomination of SAFER+ as a Candidate Algorithm for Advanced Encryption Standard (AES)- Represented at the first AES conference, Ventura, USA, August20-25, (1998)

27. P. Mohassel, S. Niksefat, S. Sadeghian,B. Sadeghiyan: An Effcient Protocol for Oblivious DFA Evaluation and Applications http://eprint.iacr.org/2011/434.pdf