Cryptography-B coursework: Paper review

Manan Vaswani

December 26, 2018

1 Introduction

At the CRYPTO2012 conference, Mike Rosulek raised an interesting question regarding black-box constructions in Cryptography, which he discussed in his paper "Must you know the code of f to securely compute f?" [1]. This paper looks at the possibility of using functions without having to compute the actual code of such a function in Multi-Party Communications.

In their 1989 paper [2], Impagliazzo and Rudich asked the important question "When do black-box constructions actually exist?". Their results showed that for random permutations ...

Black-box constructions in cryptography are those that rely only on the input and output behaviour of their components without actually knowing the details and construction of the components. They are widely used in cryptography due to the fact that they are highly practical, efficient and modular. Secure Multi-Party Computation(MPC) allows mutually distrusting parties to compute a function f on its shared inputs. One non-black box step that is used in all secure MPC communications is the evaluation of this function f. The function is expressed as a low-level circuit, and then evaluated gate by gate on the inputs provided. This means that the complexity of the communication protocol is directly dependent on the circuit complexity of the function. However, this step is unavoidable for most general purpose MPC, but the paper looks at exploring for which special purpose secure communication tasks it would be possible to have a true black box construction.

Interestingly, this topic has not been as widely researched as one would expect due to ??

2 Definitions

For a general-purpose MPC with a fixed functionality f, the protocol directly depends on f anyway, so the protocol could simply have the circuit for f hard-coded and use that every time. Instead, the author models a protocol as a pair of oracle machines that is instantiated with any functionality f that is taken from a much larger class of functionalities $\mathcal{C}$, and then emulates a functionality related to f. If this class of functionalities is particularly large, the protocol cannot construct the circuit representations of all the related functionalities. With this, he introduces the definition of a functionally-black box protocol.

$\mathcal{C}$ is a class of functions. $\mathcal{F}$ is an ideal functionality implemented as an oracle machine. A **functionally-black-box (FBB)** protocol for $\mathcal{F}^{\mathcal{C}}$ (i.e $\mathcal{F}$ instantiated with $\mathcal{C}$) is a pair of interactive oracle machines (π_A, π_B) if for all $f \in \mathcal{C}$, the protocol (π_A^f, π_B^f) is a secure protocol for the ideal functionality $\mathcal{F}^f$. This definition simply models a secure protocol that uses its functionality in a black-box way. For an FBB protocol, the adversaries may have access to an explicit representation of the function f that the protocol is instantiated with, hence there is no compromise on the security condition being observed, but the honest parties only use a black-box definition of the function. The intent of the definition above is to characterize the efficiency of the honest parties, without affecting any security conditions.

An observation is that the set $\mathcal{C}$ must not be learnable in the sense that the circuit representations of $f \in \mathcal{C}$ can be obtained by repeated interactions between the honest parties, or with an external oracle. Additionally, for all $f \in \mathcal{C}$, the domain size must be infinite, as for a constant-sized domain, $\mathcal{C}$ would be learnable by exhaustively querying the functions.

3 Proof Outline

4 Positive Example

5 Negative Example

6 Results for Malicious Security

7 Zero-Knowledge Proofs

8 Related Works

[3]

References

[1] Mike Rosulek. Must you know the code of f to securely compute f? In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages

87–104. Springer, Heidelberg, August 2012.

[2] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

[3] Yuval Ishai, Eyal Kushilevitz, Manoj Prabhakaran, Amit Sahai, and Ching-Hua Yu. Secure protocol transformations. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 430–458. Springer, Heidelberg, August 2016.