Theoretical Computer Science 651 (2016) 11–36

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Secure computation without computers ☆

Paolo D'Arco *, Roberto De Prisco *

Dipartimento di Informatica, University of Salerno, Via Giovanni Paolo II, 132, I-84084, Fisciano SA, Italy

ARTICLE INFO

Article history:
Received 25 March 2015
Received in revised form 22 June 2016
Accepted 2 August 2016
Available online 8 September 2016
Communicated by G. Ausiello

Keywords:
Yao's construction
Visual cryptography
Secure computation

ABSTRACT

The design of secure protocols which can be used *without the aid of a computer* and *without cryptographic knowledge* is an interesting and challenging research task. Indeed, protocols enjoying these features could be useful in a variety of settings where computers cannot be used or where people feel uncomfortable to interact with or trust a computer. In this paper we make a step in such a direction: we propose a novel method for performing secure two-party computations that, apart from the setup phase, requires neither a computing machinery nor cryptographic knowledge. By merging together in a suitable way two beautiful ideas of the 80's and the 90's, Yao's garbled circuit construction and Naor and Shamir's visual cryptography, respectively, we enable Alice and Bob to securely evaluate a function $f(\cdot,\cdot)$ of their inputs, x and y, through a *pure physical* process. Indeed, once Alice has prepared a set of properly constructed transparencies (for this activity a computer is useful), Bob computes the function value $f(x, y)$ by applying a sequence of simple steps which require the use of a pair of scissors, superposing transparencies, and the human visual system. Our construction builds on Kolesnikov's gate evaluation secret sharing schemes.

1. Introduction

An important goal for the crypto community is to provide secure protocols that can be used by people without computing machinery. Indeed, protocols enjoying these features could be appealing in situations where computers cannot be used or where people feel uncomfortable, for example for psychological or social reasons, to interact with or trust them. The first clear efforts towards achieving this goal can be found in papers from the end of 90's and from the beginning of the new century which deal with specific classes of problems. In [21] secure authentication and identification protocols for "unassisted humans" were proposed, and the authors explicitly stated that *"protocols which allow unaided humans to identify themselves securely and repeatedly may be feasible and should be a goal of the cryptographic community"*. The main feature of their protocols is that they are *human executable*, i.e., the user runs the protocol by performing simple computations without any technological help. Some previous papers, e.g., [33,34], had proposed visual authentication and identification schemes which require low-tech hardware items (i.e., transparencies). On the other hand, the issue of *trust in a computer* in important social tasks for masses, like voting, can be found in papers like [9], where an hybrid system for electronic voting, simple to use and preserving ballot secrecy, while improving access and robustness all at lower cost, was introduced. In all of the above examples, the focus is on designing *easy-to-use and secure protocols*. During the years several protocols for specific

☆ A preliminary version of this paper appeared with the title *Secure two-party computation: a visual way* in the Proc. of the 7th International Conference on Information Theoretic Security (ICITS 2013), November 28–30, Singapore, 2013, published by Springer in the LNCS Series, Vol. 8317, pp 18–38.

* Corresponding authors.
E-mail addresses: pdarco@unisa.it (P. D'Arco), robdep@unisa.it (R. De Prisco).

http://dx.doi.org/10.1016/j.tcs.2016.08.003
0304-3975/

tasks, meeting some of the requirements mentioned before, have been proposed. In this perspective, the problem of secure multi-party computation has not yet been considered.

Yao's Construction. Provided to the community during an oral presentation (FOCS 1986) by the author,[1] later on, it has been widely exploited in protocol design, but, apart some notable exceptions, it has more or less been considered as a powerful tool for establishing existential results. However, starting from [30], until recent years, it has been shown that fine-tuned implementations, for reasonable input sizes, are becoming practical and competitive with respect to ad hoc protocols in many settings (e.g., [22]), and thus new attention has been devoted to it. A version of the construction has been clearly described and proved secure according to precise definitions and assumptions in [29]. In a few other new recently introduced cryptographic primitives and protocols, e.g., *functional encryption* [6] or *non-interactive verifiable computing* [19], the construction plays a key role, and in [3] it has been even proposed to move from a view of Yao's construction as a cryptographic tool to a view of the construction as a *cryptographic goal*, which can be achieved with several security properties and privacy degrees.[2] From a certain point of view, Yao's idea is living nowadays a sort of *second life*.

Roughly speaking, Yao's construction, enables two parties, Alice and Bob, to privately evaluate a function $f(\cdot,\cdot)$ on their inputs, x and y, in such a way that each party gets the result and, at the same time, *preserves* the privacy of its own input, apart from what can be inferred about it by the other party from its input and the function value $f(x, y)$. For example, if the function $f(\cdot,\cdot)$ is the `xor` function, given x `xor` y and one of the input, it is impossible to preserve the other input.

In a nutshell, the construction works as follows: the function $f(\cdot,\cdot)$ is represented through a boolean circuit $C(\cdot,\cdot)$ for which, for each x, y, it holds that $C(x, y) = f(x, y)$. Yao's idea is to use the circuit as a *conceptual guide* for the computation which, instead of a sequence of `and`, `or` and `not` operations on strings of bits x and y, becomes a *sequence of decryptions* of ciphertexts. More precisely, one of the party, say Alice, given $C(\cdot,\cdot)$, computes a new object $\tilde{C}$, which is usually referred to as the *garbled circuit* [1], where:

- to each wire w of $C(\cdot,\cdot)$ are associated in $\tilde{C}$ two random keys, k_w^0 and k_w^1, which (secretly, the correspondence is not public) represent 0 and 1, and
- to each gate $G(\cdot,\cdot)$ of $C(\cdot,\cdot)$ corresponds in $\tilde{C}$ a *gate table* $\tilde{G}$ with four rows, each of which is a *double encryption*, obtained by using two different keys $k_{w_1}^a$ and $k_{w_2}^b$, for $a, b \in \{0, 1\}$, of a message which is itself a random key $k_{w_3}^c$, for $c \in \{0, 1\}$. In detail, each double encryption $Enc_{ab} = Enc_{k_{w_2}^b}(Enc_{k_{w_1}^a}(k_{w_3}^c))$ uses *one of the four* possible pairs of keys $(k_{w_1}^a, k_{w_2}^b)$, associated to the input wires (w_1, w_2) of gate $G(\cdot,\cdot)$, and the message which is encrypted is the random key $k_{w_3}^c$, associated to the wire w_3 of the output of the gate $G(\cdot,\cdot)$ *if and only if* $G(a, b) = c$. The four double encryptions $Enc_{00}, Enc_{01}, Enc_{10}$ and Enc_{11} are stored in the gate table rows in a *random* order.

Once $\tilde{C}$ has been computed, Alice sends to Bob all the gate tables $\tilde{G}$ associated to the circuit gates $G(\cdot,\cdot)$, and *reveals* the random keys k_w^0 and k_w^1, associated to all the circuit *output* wires w, and their correspondences with the values 0 and 1. Moreover, for the input wires of the circuit, she sends to Bob the random keys $k_{w_1}^{x_1}, k_{w_2}^{x_2}, \ldots, k_{w_n}^{x_n}$ corresponding to the bit-values of her own input $x = x_1 x_2 \ldots x_n$. To perform the computation represented by $\tilde{C}$, then Bob needs only the keys associated to the input wires corresponding to *his own* input, and he needs to get them without revealing his input bits to Alice. This issue is solved by means of *executions* of a 1-out-of-2 *oblivious transfer* protocol [15], through which Bob receives the random keys $k_{w_{n+1}}^{y_1}, k_{w_{n+2}}^{y_2}, \ldots, k_{w_{2n}}^{y_{2n}}$ corresponding to the bit-values of his own input $y = y_1 y_2 \ldots y_n$ while Alice does not get any information from the transfer, that is, Alice is not able to tell which specific keys Bob has recovered.

Finally Bob, according to the topology of the original circuit $C(\cdot,\cdot)$, level after level, decrypts *one and only one* entry from each gate table $\tilde{G}$ in $\tilde{C}$, until he computes *one and only one* random key associated to each output wire. Indeed, the encryption is such that only one decryption in each table is correct. The binary string which corresponds to the sequence of computed random keys, associated to the output wires, is the value $C(x, y)$. Bob sends the result of the computation to Alice.

It is easy to check that the computation is correct and, intuitively, that the privacy of the inputs is preserved. The random keys held by Bob, the rows of each $\tilde{G}$, and the random keys obtained decrypting a row in each $\tilde{G}$, do not leak any information about the actual bits of Alice's input value.

Visual Cryptography. Visual cryptography is a special type of secret sharing in which the secret is an image and the shares are random-looking images printed on transparencies. It was introduced by Naor and Shamir [34], whose model is called *deterministic* and, independently and in a different form, called *random grid*, by Kafri and Keren [24]. Later on, Yang [38] introduced a *probabilistic* model, which was generalized in [11]. However, in [14], it was proved that all these models are related to each other. In this paper we use a combination of a simple type of deterministic schemes and a random grid scheme.

[1] Latins said: *Verba volant, scripta manent*. Yao's construction disproves the saying. Indeed, [39] and [40], the papers which usually are cited when the construction is used or referred to, do not contain any description of it. It has never been written down by the author, but only provided to the community during the presentation at FOCS. Fortunately, *verba* were captured by other researchers, who used the construction and ideas of the construction in subsequent papers.

[2] The introduction of [3] offers a brief history of the construction and a nice accounting of the research efforts which followed.