Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Dual protocols for private multi-party matrix multiplication and trust computations [☆]

Jean-Guillaume Dumas [a], *Pascal Lafourcade* [b], *Jean-Baptiste Orfila* [a,*], *Maxime Puys* [c]

[a] *Université Grenoble Alpes, CNRS, LJK, UMR 5224, 700 av. centrale, IMAG/CS-40700, Grenoble 38058 CEDEX 9, France*

[b] *Université Clermont Auvergne, LIMOS, UMR 6158, Campus Universitaire des Cézeaux, BP 86, Aubière 63172 CEDEX, France*

[c] *Université Grenoble Alpes, CNRS, Verimag, UMR 5104, 700 av. Centrale, IMAG/CS-40700, Grenoble 38058CEDEX 9, France*

ARTICLE INFO

Article history:
Available online

Keywords:
Secure Multiparty Computation (MPC)
Distributed Matrix Multiplication
Trust evaluation
Proverif
AVISPA

ABSTRACT

This paper deals with distributed matrix multiplication. Each player owns only one row of both matrices and wishes to learn about one distinct row of the product matrix, without revealing its input to the other players. We first improve on a weighted average protocol, in order to securely compute a dot-product with a quadratic volume of communications and linear number of rounds. We also propose two dual protocols with five communication rounds, using a Paillier-like underlying homomorphic public key cryptosystem, which is secure in the semi-honest model or secure with high probability in the malicious adversary model. Using cryptographic protocol verification tools, we are able to check the security of both protocols and provide a countermeasure for each attack found by the tools. We also give a generic randomization method to avoid collusion attacks. As an application, we show that this protocol enables a distributed and secure evaluation of trust relationships in a network, for a large class of trust evaluation schemes.

1. Introduction

Secure multiparty computations (MPC), introduced by Yao (Yao, 1982) with the millionaires‘ problem, have been intensively studied during the past thirty years. The idea of MPC is to allow n players to jointly compute a function f using their private inputs without revealing them. In the end, they only know the result of the computation and no more information. Depending on possible corruptions of players, one may prove that a protocol may resist against a collusion of many players, or that it is secure even if attackers try to maliciously modify their inputs. Mostly any function can be securely computed (Ben-Or et al., 1988) and many tools exist to realize MPC protocols. They comprise for instance the use of a Trusted Third Party (Du and Zhan, 2002), the use of Shamir's secret sharing scheme (Shamir, 1979), or more recently the use of homomorphic encryption (Goethals et al., 2005). It is also possible to mix these techniques (Damgård et al., 2012).

[☆] This work was partially supported by "Digital trust" Chair from the University of Auvergne Foundation, by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025) and by the OpenDreamKit 2020 European Research Infrastructures project (#676541).

[*] *Corresponding author.*
E-mail addresses: jean-baptiste.orfila@univ-grenoble-alpes.fr (J.-B. Orfila), jean-guillaume.dumas@univ-grenoble-alpes.fr (J.-G. Dumas), pascal.lafourcade@uca.fr (P. Lafourcade), maxime.puys@univ-grenoble-alpes.fr (M. Puys).

http://dx.doi.org/10.1016/j.cose.2017.04.013
0167-4048/© 2017 Elsevier Ltd. All rights reserved.

Our goal is to apply MPC to the distributed evaluation of trust, as defined in Jøsang (2007). Indeed, there are several schemes for evaluating the transitive trust in a network. Some use a single value representing the probability that the expected action will happen; the complementary probability being an uncertainty on the trust. Others include the *distrust* degree indicating the probability that the opposite of the expected action will happen (Guha et al., 2004). More complete schemes can be introduced to evaluate trust: Jøsang introduces the *Subjective Logic* notion which expresses beliefs about the truth of propositions with degrees of "uncertainty" in Jøsang (2007). In *e.g.* Foley et al. (2010) algorithms are proposed to quantify the trust relationship between two entities in a network, using transitivity and reachability. Then the authors Huang and Nicol (2010) applied the associated calculus of trust to public key infrastructures. For instance, in Dumas and Hossayni (2012), aggregation of trusts between players on a network is done by a matrix product defined on two monoids (one for the addition of trust, the other one for multiplication, or transitivity): each player knows one row of the matrix, its partial trust on its neighbors, and the network as a whole has to compute a distributed matrix squaring. Considering that the trust of each player for his colleagues is private, at the end of the computation, nothing but one row of the global trust has to be learned by each player (*i.e.,* nothing about private inputs should be revealed to others). Thus, an MPC protocol to resolve this problem should combine privacy (nothing is learned but the output), safety (computation of the function does not reveal anything about inputs) and efficiency (Lindell, 2009). First, we need to define an MPC protocol which allows us to efficiently compute a distributed matrix product with this division of data between players. The problem is reduced to the computation of a dot product between vectors U and V such that one player knows U and V is divided between all players.

1.1. *Related work*

Dot product in the MPC model has been widely studied (Amirbekyan and Estivill-Castro, 2007; Du and Atallah, 2001; Wang et al., 2008). However, in these papers, assumptions made on data partitions are different: there, each player owns a complete vector, and the dot product is computed between two players where in our setting, trust evaluation should be done among peers, like certification authorities. For instance, using a trusted third party or permuting the coefficients is unrealistic. Then, to reduce some communication costs of sharing schemes, the use of homomorphic encryption is natural (Goethals et al., 2005). In such cases the underlying homomorphic cryptosystem is quite often that of Paillier (Paillier, 1999) or of Benaloh (Benaloh, 1994; Fousse et al., 2011).

Now, computing a dot product with n players is actually close to the MPWP protocol of Dolev et al. (2010), computing a mean in a distributed manner: computing dot products is actually similar to computing a weighted average where the weights are in the known row, and the values to be averaged are privately distributed. The original MPWP protocol has to use a Benaloh-like homomorphic encryption. We here show how to use instead a Paillier-like system, usually faster in practice. Also, in MPWP the total volume of communication for a dot product is $\mathcal{O}(n^3)$ with $\mathcal{O}(n)$ communication rounds. Intuitively, by making the first player masking each term in the dot-product sum, instead of having all the players mask their own, we can get rid of the secret sharing part of MPWP. This enables us first to reduce the amount of communications from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ and, second, to reduce the number of communication rounds. Indeed instead of having $\mathcal{O}(n)$ steps, we can use a setting with $\lfloor\frac{n-1}{2}\rceil$ parallel rounds, each one with less than five parallel communication steps. Thus, we propose in Section 6 a way to reduce the parallel complexity of the protocol (that could also apply to MPWP): we replace a ring of n players into parallel rings of 3 or 4. Security is modified, as the new protocol is secure against semi-honest adversaries while the initial one was secure against a coalition of malicious ones. Now we also propose a generic mitigation scheme, which we call a *random ring order*, that allows to recover the security against malicious adversaries, with high-probability, while preserving efficiency. We then obtain a first secure and efficient protocol.

Another possibility is to use a two-phase protocol sketched in Yao et al. (2007): this $\mathcal{O}(n)$ protocol requires to homomorphically exchange vector coefficients in a first phase. Then, it uses a multiparty summation protocol. In Yao et al. (2007) it is suggested to use protocols by Atallah and Du (2001) for the summation in the second phase. We show here that this summation protocol is not resistant against a coalition of malicious insiders. To repair the protocol, one can use instead a secret sharing scheme, but this is back to an $\mathcal{O}(n^2)$ communication protocol. We here instead propose first to use Paillier-like homomorphic schemes within the whole protocol. Second we prove that it is possible to use a classical salary-sum protocol for the summation phase, thanks to our novel random ring order mitigation scheme. We thus also preserve both advantages, a $\mathcal{O}(n)$ time and communications costs as well as security against malicious adversaries. This resulting protocol is then actually somewhat dual to our first one.

Note that several other generic MPC protocols exist, usually evaluating circuits, but they require $\mathcal{O}(n^3)$ computations and/or communications per dot-product (Bendlin et al., 2011; Damgård et al., 2012).

1.2. *Contributions*

Overall, we provide the following results:

1. A fully secure protocol, *P-MPWP*, improving on *MPWP*, which reduces both the computational cost, by allowing the use of Paillier's cryptosystem, and the communication cost, from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$.
2. An $\mathcal{O}(n)$ time and communications protocol, called *Distributed and Secure Dot-Product* ($DSDP_n$) (for n participants), which allows us to securely compute a dot product UV, against a semi-honest adversary, where one player, the master, owns a vector U and where each player knows one coefficient of V.
3. A parallel variant that performs the dot-product computation in parallel among the players and thus uses a constant total number of rounds. This is extended to a *Parallel Distributed and Secure Matrix-Multiplication* ($PDSMM_l$) family of protocols.

Please cite this article in press as: Jean-Guillaume Dumas, Pascal Lafourcade, Jean-Baptiste Orfila, Maxime Puys, Dual protocols for private multi-party matrix multiplication and trust computations ☆, computers & security (2017), doi: 10.1016/j.cose.2017.04.013