Poster: Distributed Delegation of Computation with Verification Outsourcing

Gang Xu, George Amariucai, and Yong Guan*
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011
Email: {gxu, gamari, guan}@iastate.edu

I. Motivation and Problem Statement

Cloud computing is bound to become the leading trend of modern computing. Its potential client base is extremely diverse, ranging from small businesses, trying to cut down on their computation and storage costs, to private users trying to run computation-intensive applications on their lightweight, hand-held devices, and to the military, trying to opportunistically employ both trusted and untrustworthy computational resources for quick information processing, leading to responsible decision making. In the *delegation of computation* (or *computation outsourcing*) paradigm, the client delegates a computational task to the server. The client provides the server with the input of the computational task. The server produces a result, and returns it to the client. Should the client require result assurance, he can start a standard verification protocol, where the server and the client assume the roles of the *prover* and the *verifier*, respectively. The problem of *delegation of computation* has been intensely investigated, and almost-practical verification algorithms have been recently proposed in [1] [2] [3] [4]. Moreover, the idea of *delegation of verification*, first introduced in [5], is bound to additionally decrease the computational costs of the client, by outsourcing verification to a third party. Of course, new soundness and confidentiality issues arise in this context.

Yet another, more challenging application of cloud computing emerges in distributed environments, and is the focus of the current research project. While civilian applications abound, for demonstration purposes we like to refer to a more security-sensitive example, like the one presented by a combat environment.

Imagine a soldier squad, or a squadron of aircraft deployed in the field, acquiring, sharing and processing information to support decision making. Since information and (as a consequence) computational loads may easily become overwhelming (especially for a single one of the lightweight devices carried by infantry troops), the need for distributed delegation of computation becomes apparent. In this scenario, the squad (or squadron) leader plays the role of the *client*, while all other available computation resources play the role of the *server*. The *available computation resources* include, but are not limited to, the computing devices carried by the squad (or squadron) members. However, in addition to these, the local infrastructure may be used to help the computation. This approach immediately implies a need for confidentiality. Moreover, since all protocols are negotiated over a wireless medium, the confidentiality constraints on the delegation of computation would also protect against undetected intrusion by malicious devices.

A basic scenario is depicted in Figure 1, where squad members, the local infrastructure, and enemy soldiers are all part of two protocols: delegation of computation, and delegation of verification. The local infrastructure may prove to be a computational asset, but may not be trusted with sensitive information, while the enemy soldier will probably attempt to interfere with the correctness of the protocol, or intelligently influence the protocol in a way that leads to maximum information leakage. Consequently security mechanisms have to be implemented, to restrict the amount of information that leaks to the delegates about: (a) the details of the computational task, (b) the input of the computational task and (c) the result of the computational task. In addition, since verification is also outsourced in a distributed manner, an additional layer of security mechanisms should ensure that the verification protocol maintains its soundness, even in the presence of colluding cheating or lazy verifiers.

II. Methods and the Key Contribution

We aim to combine the distributed delegation of computation with confidentiality constraints, and the distributed delegation of verification, such that the set of provers coincides with the set of verifiers. In fact this new framework implies only two types of actors: the client $\mathcal{C}$ and the multiple prover/verifiers $\mathcal{P}/\mathcal{V}$. We aim at keeping the computation and/or the inputs/outputs (at least partially) confidential from the prover/verifiers. Intelligent mixing and scheduling of the delegation and verification tasks is required to maintain the soundness of the protocol. Intelligent mixing and scheduling would decrease the correlation between the computation and verification tasks at each prover/verifier. This would reduce the prover/verifier's ability to cheat during the computation and verification processes, while ensuring that the prover/verifier cannot recover too much information about the computation task, its inputs and its outputs. In essence, each compu-

* Gang is currently a Ph.D student, while Dr.Guan and Dr.Amariucai are faculties at ECE department in Iowa State University.

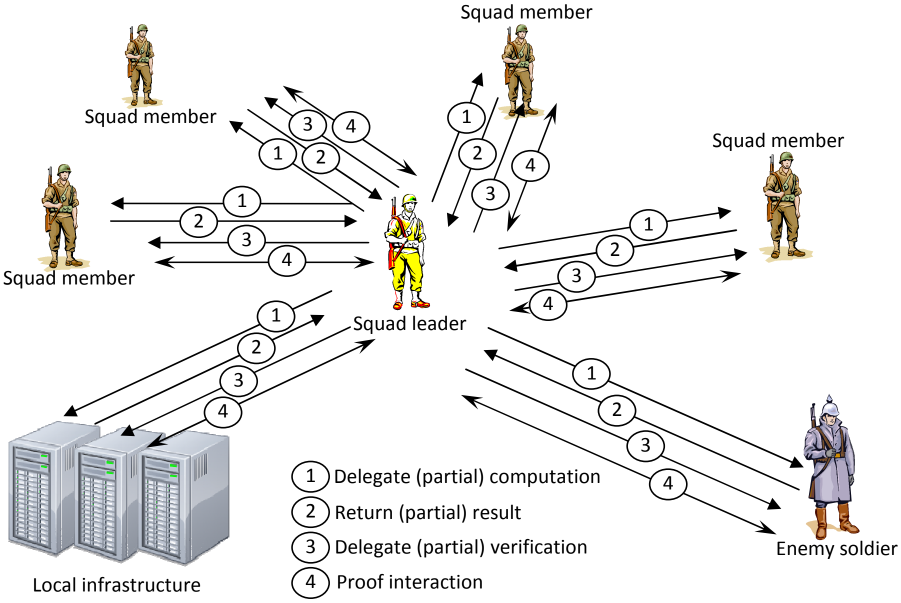

Fig. 1. A basic scenario of distributed delegation of verification with distributed delegation of verification.

tation/verification assignment is masked by other computation/verification assignments.

The challenge is to ensure that such exposure, which is beyond the control of the client, does not compromise the confidentiality and soundness of the protocol. Information leakage may prove to be inevitable under particular network conditions. For these cases, we design a mechanism that can easily detect and isolate the point where such information leakage occurs. The following cases shall be considered under this topology: (1) the client has the full knowledge of the topology of the network of all verifiers/provers; (2) the client has partial or no knowledge of the topology.

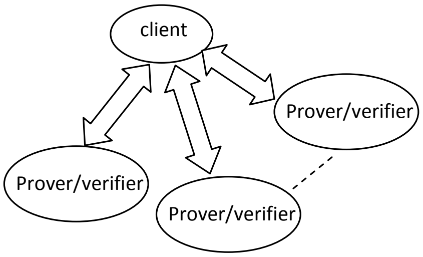

Fig. 2. Delegation of computation with delegation of verification in a simple star topology.

The first case requires a deeper understanding of the network topology. We shall decompose the network according to graph theory and find out critical nodes along the verification/computation chain. Critical nodes are those that are exposed to the most information about the computational task and its inputs/outputs. Special algorithms will be required when interacting with the critical nodes. For the second case, a tradeoff between security constraints and efficiency will be provided.

To summarize, the contributions of our research project can be stated as follows:

1) We propose to augment the distributed-delegation-of-computation paradigm with distributed delegation of verification. While the confidentiality aspect of delegated verification can be solved by an extension of the results in [5], we introduce more efficient algorithms, that rely on intelligently mixing the computation results and verification tasks during the verification-delegation process.
2) We demonstrate, for the first time, that in a distributed environment, the same computational resources used for performing the delegated computation can also be used for performing the verification, in a secure and sound manner. We focus on a randomly-connected topology with multiple, collaborating delegation and aggregation nodes.

REFERENCES

[1] S. Setty, A. J. Blumberg, and M. Walfish, "Toward practical and unconditional verification of remote computations," in *Proceedings of the 13th USENIX conference on Hot topics in operating systems*, ser. HotOS'13. Berkeley, CA, USA: USENIX Association, 2011, pp. 29–29.

[2] S. Setty, R. McPherson, A. J. Blumberg, and M. Walfish, "Making argument systems for outsourced computation practical (sometimes)," in *NDSS*, 2012.

[3] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct nizks without pcps," in *Proceedings of the IACR Eurocrypt Conference*, ser. Eurocrypt 13, 2013.

[4] B. Parno, C. Gentry, J. Howell, and M. Raykova, "Pinocchio: nearly practical verifiable computation," in *the IEEE Symposium on Security and Privacy*, ser. IEEE S&P 13, 2013.

[5] G. Xu, G. Amariucai, and Y. Guan, "Delegation of computation with verification outsourcing: Curious verifiers," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, ser. PODC '13. ACM, 2013.