Secured Outsourced Linear Algebra

Amrit Kumar, Jean-Louis Roch, Clément Pernet

Secured Outsourced Linear Algebra

Amrit Kumar
Department of Computer Science
École polytechnique
Palaiseau, France
email: amrit.kumar@polytechnique.edu

Jean-Louis Roch
MOAIS, LIG-INRIA joint team
University of Grenoble
Grenoble, France
email: jean-louis.roch@imag.fr

Clément Pernet
MOAIS, LIG-INRIA joint team
University of Grenoble
Grenoble, France
email: clement.pernet@imag.fr

***Abstract*—We propose an interactive algorithmic scheme for outsourcing matrix computations on untrusted global computing infrastructures such as clouds or volunteer peer-to-peer platforms. In this scheme, the client outsources part of the computation with guaranties on both the inputs' secrecy and output's integrity. For the sake of efficiency, thanks to interaction, the number of operations performed by the client is almost linear in the input/output size, while the number of outsourced operations is of the order of matrix multiplication. The scheme is homomorphic, based on linear codes (especially evaluation/interpolation version of Reed-Solomon codes). Privacy is ensured by encoding the inputs using a secret generator matrix, while fault tolerance is ensured by the high error detection and correction capability of the code. The scheme can tolerate multiple malicious errors and hence provides an efficient solution beyond resilience against soft errors.**

I. Scenario

Computational power is often asymmetric. On the one hand, global computing platforms such as clouds available through internet provide a large scale of computational power and resources, but remain susceptible to various kinds of malicious attacks and faults. While on the other hand, a relatively weak local client is secure and reliable. The goal of today's computing infrastructure is to provide solution to draw the benefits of both of these components while ensuring secrecy of certain inputs.

In a scenario (Figure 1) where a weak but reliable client outsources a computation on a given input, the client should be able to efficiently verify the correctness of the result returned by the untrusted platform. Solutions for general computations either rely on Probabilistically Checkable Proofs (PCPs) [1], or fully-homomorphic encryption (FHE) schemes as in [5]. Considering their complexity, these constructions are currently beyond practical use. However, authors in [9] propose a new construction that can efficiently verify general computations under cryptographic assumptions. This construction compactly encodes computations as quadratic programs [4] which are then encoded as elements of a group equipped with a bilinear map. The weak client receives a proof (of constant size) along with the computational result. The verification procedure involves group operations.

Furthermore, on large scale computing systems error resilience is an issue. Error probability increases with the node count [2]. Algorithm-based Fault Tolerance (ABFT) [7] solutions have been explored for matrix computations without considering privacy. Focusing on a small rate of soft errors, an ABFT dense linear system solver is provided in [2] that is based on a low density parity check. Soft errors in general are produced in case the computing system is subject to cosmic radiations. Yet, on externalized computing platforms, malicious attacks that may corrupt a large number of intermediate computations are of concern. Such massive attacks occur due to Trojan attacks, and more generally orchestrated attacks against widespread vulnerabilities of a specific operating system that may result in the corruption of a large number of resources. In [10], an ABFT efficient solution for matrix multiplication is proposed for integrity against massive attacks that is based on a Reed-Solomon code.

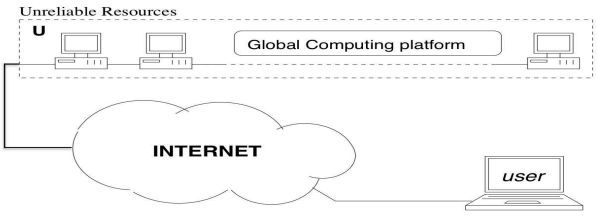

Fig. 1. Outsourcing a task to an untrusted platform

In this work we extend this solution in both directions. First, to provide secrecy, we use a secret code, based on a private Vandermonde generator matrix. Second, we extend the scheme to more general matrix computations such as LU factorization or matrix powering (with application to connected components in a graph), where the output data is non-linear in the input data. A major constraint is efficiency: trivial lower bounds for the cost of an interactive scheme are the size of the input and the output (memory cost) and the work of the best known algorithm (computational cost). We propose an asymmetric scheme for matrix multiplication that is almost optimal with respect to both the input/output size on the reliable resource (user side) and the best upper bound on the unreliable one (global computing platform side).

II. ABFT Dense Matrix Multiplication

As most of the linear algebra computations reduce to dense matrix multiplication, the design of the interactive zero knowledge protocol for computations is based on outsourcing matrix product. For the sake of clarity, we restrict in the sequel to $k \times k$ square matrices. The goal of these protocols is to keep the complexity of the operations almost linear in the size $O(k^2)$ of the input on the weak client, while on the unreliable cloud, a complexity $\widetilde{O}(n^\omega)$ would be acceptable,

This work has been partially supported by the LabEx PERSYVAL-Lab: n°. ANR-11-LABX-0025.

where $2 < \omega \leq 3$ denotes the exponent of matrix multiplication cost. A standard way for ABFT matrix multiplication consists in encoding the left and right operands by multiplying each by the generator matrix of a linear code [2], [7]. In the sequel, we use Evaluation-Interpolation linear codes, denoted RS (Reed-Solomon). These codes defined over a base field $\mathbb{F}$ are maximum distance separable. Assuming $\mathbb{F}$ larger enough, for any n with $\text{card}(\mathbb{F}) \geq n > k$, an (n, k) RS code is characterized by a $k \times n$ matrix G. A source vector x of size k is encoded by $y = x \cdot G$; any configuration of $(n-k)/2$ errors in y is guaranteed to be corrected.

The proposed secured ABFT protocol is as follows. The weak client initiates the protocol by generating two (n, k) RS codes, defined by G_1 and G_2. The input $k \times k$ matrices A and B are encoded as: $G_A = tr(G_1) \cdot A$ and $G_B = B \cdot G_2$, where tr denotes the transpose map. G_1 and G_2 are kept secret which eventually makes A and B secret. The client sends G_A and G_B to the global platform that performs the computation and sends back a result matrix R. Various errors may occur during computations or communications that are modeled by an insecure or noisy channel: the received matrix is seen as a perturbation of the correct encoded result $G_C = G_A \cdot G_B$. Upon decoding R, the client obtains a matrix $\widetilde{C}$ which verifies $\widetilde{C} = A \cdot B + E$, where $E_{n \times n}$ is the error matrix. The client can correct up to $(n-k)^2/4$ errors in $\widetilde{C}$ to recover $C = A \cdot B$.

The cost of computation on the reliable client sums to the cost of encoding and decoding. The encoding of each row or column reduces to k polynomial evaluations of degree k in n points, each computed in $O(n\log^2 n)$ with precomputation, so $\widetilde{O}(n^2)$ for the full matrices A and B. With fast extended GCD, decoding can be performed in $O(n\log^2 n)$ for each row or column, so $\widetilde{O}(n^2)$ for the matrix $\widetilde{C}$. The multiplication is performed by the remote platform in $O(n^\omega)$.

The client also has the possibility to verify the correctness of the result. This verification allows to detect cheating workers and even prevents man-in-the-middle (MITM) attacks. A cheating worker may not compute the matrix product correctly (and hopes that the fault remains undetected) while in the MITM scenario, an adversary might simply intercept the communication between the client and the platform and replaces the result by some other *good-looking* matrix, ex. the adversary might replace the matrix R by $G_A tr(G_A)$. For verification i.e. testing if $C = AB$, we propose to use probabilistic Freivalds' algorithm [3] which runs in $O(n^2)$, and states the correctness with good probability.

To ensure secrecy of inputs, the generator matrices G_1 and G_2 are kept secret. The evaluation points are randomly chosen and kept secret. However, if secrecy is discarded, the evaluation points are chosen to be $1, \alpha, \alpha^2, \ldots, \alpha^{\text{card}(\mathbb{F})-2}$, where α is a generator of the multiplicative group of the base field $\mathbb{F}$. With this choice of evaluation points, Fast Fourier Transformation (FFT) allows fast encoding and decoding and provides a logarithmic advantage. We note that a small field (such as $\mathbb{F}_2$) would not provide enough evaluation points.

III. Illustrative Example : Interactive Block LU Decomposition

Extending the idea, we also propose an ABFT interactive block LU decomposition protocol where the client outsources the task of LU decomposition to the platform. This protocol follows the standard block LU decomposition algorithm, where the inverse of the diagonal element is calculated locally and the blocks below the diagonal element are updated. The task of updating the sub-matrix to the right and below of the diagonal element is shared between the cloud and the client. We note that sub-matrix update requires the computation of matrix multiplication, hence is outsourced thanks to the previous matrix multiplication. Upon retrieving the product, the update operation reduces to addition and is performed locally. This interaction outsources the larger part of computation to the remote platform, while the smaller part is performed by the client. Let K be the block size: the cost of computation on client's side is $\widetilde{O}(n^2 K^{\omega-2})$ for block inversions and column update, and $\widetilde{O}(n^\omega K^{-\omega+2})$ for sub-matrix updates; while on the untrusted cloud, the cost is $O(n^\omega)$. For $\omega = 3$, the optimality is obtained when the block size is $K = \sqrt{n}$.

IV. Conclusion

In this paper, we design an efficient alternative to FHE based outsourcing with acceptable practicality and security. Our ABFT solution is resilient against malicious errors and hence goes beyond the correction of soft errors and can even handle MITM attacks. The scheme computes matrix operations such as matrix-matrix multiplication and can be extended to interactive protocols performing more complex operations on matrices. The ongoing work includes quantifying security provided by our scheme, the study of the related cost-security trade-off and include other computations in the framework. While this work is based on large finite fields, a perspective is the design of efficient solution for floating point numbers, based on dedicated encoding for matrix multiplication.

References

[1] Sanjeev Arora. Probabilistic checking of proofs: a new characterization of np. In *Journal of the ACM*, pages 2–13, 1998.

[2] Peng Du, Piotr Luszczek, and Jack Dongarra. High performance dense linear system solver with resilience to multiple soft errors. In *ICCS*, pages 216–225, 2012.

[3] Rusins Freivalds. Probabilistic machines can use less running time. In *IFIP Congress*, pages 839–842, 1977.

[4] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps.

[5] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[6] Philippe Golle and Ilya Mironov. Uncheatable distributed computations. In *Lecture Notes in Computer Science*, pages 425–441. Springer, 2001.

[7] Kuang-Hua Huang and J. A. Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE Trans. Comput.*, 33(6):518–528, June 1984.

[8] Fabian Monrose, Peter Wycko, and Aviel D. Rubin. Distributed execution with remote audit. In *In Proceedings of the 1999 ISOC Network and Distributed System Security Symposium*, pages 103–113, 1999.

[9] Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. Cryptology ePrint Archive, Report 2013/279, 2013. http://eprint.iacr.org/.

[10] Jean-Louis Roch and Sebastien Varrette. Probabilistic certification of divide & conquer algorithms on global computing platforms. application to fault-tolerant exact matrix-vector product. In ACM publishing, editor, *Parallel Symbolic Computation'07 (PASCO'07)*, London, Ontario, Canada, July 2007.

[11] Radu Sion. Query execution assurance for outsourced databases, 2005.