**Fifty-Fifth Annual Allerton Conference**
**Allerton House, UIUC, Illinois, USA**
**October 3-6, 2017**

# Privacy preserving Cloud-based Quadratic Optimization

Andreea B. Alexandru Konstantinos Gatsis George J. Pappas

***Abstract*— This work proposes a protocol for privately solving constrained quadratic optimization problems with sensitive data. The problem encompasses the private data of multiple agents and is outsourced to an untrusted server. We describe the desired security goals and investigate the information leakage from duality theory. We present an interactive protocol that achieves the solution of a strictly quadratic convex optimization problem with private linear cost and private linear inequality constraints, by making use of partially homomorphic cryptosystems to securely effectuate computations. Then, we provide extensions to the protocol in order to also consider equality constraints and to obtain a speedup of the performance.**

## I. Introduction

In the Internet of Things setup, cloud-outsourced computation is ubiquitous, due to the low computation, battery and storage requirements of the devices. The cloud aggregates the users' data, effectuates a complex computation on it and obtains the desired result. Due to the increasing number of cyberattacks, privacy infringements and financial interests arising from owning private data [1], [2], [3], it is unrealistic to assume that the cloud does not try to take advantage of the users' data. The most common framework in which multiparty computation is performed is the semi-honest model, which, intuitively, describes rival parties that collaborate to achieve a common goal. Under this setup, we wish to develop protocols that satisfy cryptographic security, i.e., no party can infer anything about the private data of other parties.

### A. Related work

The research topic of secure cloud-based computing has been very active in the past decades, and several tools have been proposed, such as garbled circuits, homomorphic encryption, oblivious transfer, secret sharing, differential privacy [4], [5], [6], [7] etc. Using cryptographic tools provides strong security guarantees but inevitably increases the computational and communication complexity. When dealing with algorithms that necessitate a large number of iterations, it is desirable to choose a cryptographic method that guarantees security with a low cost of complexity, and, furthermore, that can be efficiently particularized to the current problem. Both fully homomorphic encryption and garbled circuits offer frameworks for general computations, but at the cost of a very high complexity and substantial depth [8], [9]. On the other hand, differential privacy mechanisms lead to loss of accuracy in the solution, prompting a compromise between the level of privacy and the performance of the optimization algorithm, e.g. [10], [11].

Several gradient descent methods have been proposed in [12], [13] for unconstrained optimization problems in machine learning and data mining, under different cryptographic setups, but the biggest challenges in the optimization algorithms are represented by the projection on a feasible space, which they do not address. Secure simplex algorithms for linear programs have been proposed in [14], [15], however, the simplex method does not scale well with more general optimization problems. The problem of secure constrained quadratic optimization under the setup of additively homomorphic encryption schemes was addressed in [16], but their proposed protocol revealed sensitive information and guaranteed only conditional privacy.

Due to the above reasons regarding complexity, for the problem of outsourced constrained quadratic optimization with private data, we will not address schemes that are fully homomorphic or employ garbled circuits. Instead, we will focus on partially homomorphic encryption schemes, specifically additively homomorphic encryption schemes, which have significantly less overhead than their fully counterpart.

### B. Problem formulation

In control theoretic problems (and not only), it is very common that linear and quadratic optimization problems arise, e.g., state estimation under minimum square error, Model Predictive Control, Support Vector Machines etc. We consider the following setup in which a quadratic optimization problem with distributed private data is solved. There are three types of parties involved in the problem: a number of agents $\mathcal{A}_i$, $i = 1, \dots, p$, a cloud $\mathcal{C}$ and a target node $\mathcal{T}$. The purpose of this setup is to compute an optimization problem with the data from the agents $\mathcal{A}_i$ and the algorithm from the cloud $\mathcal{C}$, and send the result to the target node $\mathcal{T}$. Let us consider a strictly-convex quadratic optimization problem:

$$\begin{aligned} x^* \;=\; & \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \; \frac{1}{2}x^\intercal Q_{\mathcal{C}} x + c_{\mathcal{A}}^\intercal x \\ & s.t.\; A_{\mathcal{C}} x \preceq b_{\mathcal{A}} \end{aligned} \tag{1}$$

The parties are described as follows:

**Agents** $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_p)$: The agents are semi-honest parties that possess the private information $b_{\mathcal{A}}$ and $c_{\mathcal{A}}$. The private information is decomposed across the agents in the following way: $b_{\mathcal{A}} = (b_1, \dots, b_p)$ and $c_{\mathcal{A}} = (c_1, \dots, c_p)$, with $b_i \in \mathbb{R}^{m_i}$ and $c_i \in \mathbb{R}^{n_i}$ such that $\sum_{i=1}^p m_i = m$ and $\sum_{i=1}^p n_i = n$.

**Cloud** $\mathcal{C}$: The cloud is a semi-honest party that possesses the private information $Q_{\mathcal{C}} \in \mathbb{S}^n_{++}$ and $A_{\mathcal{C}} \in \mathbb{R}^{m\times n}$. This

This work was supported by NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy.

The authors are with the Department of Electrical of Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104. {aandreea, kgatsis, pappasg}@seas.upenn.edu.

captures the case when the computation involves proprietary algorithms. In order to capture a greater number of problems, we will also consider the case where $Q_{\mathcal{C}}$ or $A_{\mathcal{C}}$ are public data, as explained in Remark 3.

**Target Node** $\mathcal{T}$: The target node is a semi-honest party that has a known public key $pk_{\mathcal{T}}$ and a secret key $sk_{\mathcal{T}}$ for the Paillier cryptosystem [17], which we will describe in Section IV-A. The target node is supposed to receive the result $x^*$ of the computation carried out at the cloud $\mathcal{C}$.

This problem can be addressed by homomorphic encryption schemes, where, in short, addition and multiplication commute with the encryption function. Thus, a party can process encrypted data without having direct access to the data. Homomorphic encryption schemes can be fully homomorphic (proposed in [18]), partially homomorphic (e.g. [17], [19]) or 'somewhat' homomorphic (e.g. [20]). However, fully and somewhat homomorphic encryption schemes have high complexity and execution times, so we will focus on partially homomorphic encryption schemes, more specifically, additively homomorphic encryption schemes.

We propose a projected gradient ascent method for the dual problem and show it is compatible with an additively homomorphic encryption scheme. We choose this method over more elaborate ones, such as interior point methods, that require operations such as inversion and multiplication between the encrypted data and can only be achieved with more complex schemes.

The rest of the paper is organized as follows: we present the terminology, along with the privacy goals in Section II. We discuss the possible information leakage of the gradient method in Section III. Section IV assembles the protocol by discussing the secure subroutines that compose it, while in Section V, the security claims are proved, by resorting to cryptographic assumptions and further theoretical notions. In Section VI, we discuss the extension on the previous algorithm to include equality constraints. The conclusions of this work and future directions of research are discussed in Section VII.

## II. Terminology and security goals

The parties in the setup we consider are semi-honest, which is a concept defined as follows:

*Definition 1:* (Semi-honest) A party is semi-honest if it correctly follows the steps of the protocol but it may store the transcript of the messages exchanged and try to learn more information than allowed by processing the data received.

We introduce some cryptographic notions that are substantial to the security definitions. In what follows, $\{0,1\}^*$ defines a sequence of bits of unspecified length. A circuit $C$ is a directed acyclic graph with internal nodes marked by gates of and, or, negation. A polynomial-size circuit family is an infinite sequence of Boolean circuits such that, for every $n$, the circuit $C_n$ has $n$ input nodes and size $p(n)$, for a polynomial $p(\cdot)$. We denote by $t \leftarrow X$ an element $t$ sampled from a probability distribution $X$. An ensemble indexed by a bit string $X = \{X_w\}_{w\in\{0,1\}^*}$ is a sequence of random variables ranging over strings of length polynomial in $|w|$. Essentially, two ensembles are called computationally indistinguishable if no efficient algorithm can distinguish between them.

*Definition 2:* (Computational Indistinguishability [21, Ch. 3]) The ensembles $X = \{X_w\}_{w\in S}$ and $Y = \{Y_w\}_{w\in S}$ are **computationally indistinguishable**, denoted $\overset{c}{\equiv}$ if for every polynomial-size circuit family $\{C_n\}_{n\in\mathbb{N}}$, every positive polynomial $p$, every sufficiently large $n$ and every $w \in S$, the following holds:

$$|\Pr_{t\leftarrow X_w}[C_n(t) = 1] - \Pr_{t\leftarrow Y_w}[C_n(t) = 1]| < 1/p(n).$$

Further, we define the privacy goals that the protocol we design should guarantee. We require **two-party privacy** of the sensitive data of the parties and **multi-party privacy**, respectively. The intuition behind these definitions is that a secure protocol does not leak any more information than what can be obtained solely from its inputs and outputs.

*Definition 3:* (Two-party privacy w.r.t. semi-honest behavior [22, Ch. 7]) Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a functionality, and $f_1(x_1, x_2)$, $f_2(x_1, x_2)$ denote the first and second element of $f(x_1, x_2)$, for any inputs $x_1, x_2 \in \{0,1\}^*$. Let $\Pi$ be a two-party protocol for computing $f$. The **view** of the $i$-th party ($i = 1, 2$) during an execution of $\Pi$ on the inputs $(x_1, x_2)$, denoted $V_i^{\Pi}(x_1, x_2)$, is $(x_i, coins, m_1, \ldots, m_t)$, where $coins$ represents the outcome of the $i$'th party's internal coin tosses, and $m_j$ represents the $j$-th message it has received. For a deterministic functionality $f$, we say that $\Pi$ **privately computes** $f$ if there exist probabilistic polynomial-time (ppt) algorithms, called **simulators**, denoted $S_i$, such that:

$$\{S_i(x_i, f(x_1, x_2))\}_{x_{1,2}\in\{0,1\}^*} \overset{c}{\equiv} \{V_i^{\Pi}(x_1, x_2)\}_{x_{1,2}\in\{0,1\}^*}.$$

*Definition 4:* (Multi-party privacy w.r.t. semi-honest behavior [22, Ch. 7]) Let $f : (\{0,1\}^*)^s \to (\{0,1\}^*)^s$ be a $s$-ary functionality, where $f_i(x_1, \ldots, x_s)$ denotes the $i$-th element of $f(x_1, \ldots, x_s)$. Denote the inputs by $\bar{x} = (x_1, \ldots, x_s)$. For $I = \{i_1, \ldots, i_t\} \subset [s] = \{1, \ldots, s\}$, we let $f_I(\bar{x})$ denote the subsequence $f_{i_1}(\bar{x}), \ldots, f_{i_t}(\bar{x})$, which models a coalition of a number of parties. Let $\Pi$ be a $s$-party protocol that computes $f$. The **view** of the $i$-th party during an execution of $\Pi$ on the inputs $\bar{x}$, denoted $V_i^{\Pi}(\bar{x})$, is defined as in Definition 3, and we let the view of a coalition be denoted by $V_I^{\Pi}(\bar{x}) = (I, V_{i_1}^{\Pi}(\bar{x}), \ldots, V_{i_t}^{\Pi}(\bar{x}))$. For a deterministic functionality $f$, we say that $\Pi$ **privately computes** $f$ if there exist simulators $S$, such that, for every $I \subset [s]$, it holds that, for $\bar{x}_t = (x_{i_1}, \ldots, x_{i_t})$:

$$\{S(I, (\bar{x}_t), f(\bar{x}_t))\}_{\bar{x}\in(\{0,1\}^*)^s} \overset{c}{\equiv} \{V_I^{\Pi}(\bar{x})\}_{\bar{x}\in(\{0,1\}^*)^s}.$$

The intuition behind the above definitions can also be viewed as follows: consider a black-box protocol that receives the inputs of Problem (1) and solves it, not exchanging any messages in order to achieve the solution. In other words, meeting the requirements of the above definitions is equivalent to not revealing any other information than what is already known to any of the parties, i.e. inputs, prescribed outputs, if any, and previously known side information, meaning details about the optimization algorithm.

*Remark 1:* Satisfying Definitions 3 and 4 is a stronger requirement than guaranteeing that an adversary cannot uniquely retrieve the data of the honest parties.

Revealing sensitive information does not always lead to a unique retrieval of the private data. Nevertheless, any piece of information revealed by the protocol, that cannot be obtained only from its inputs and outputs, leads to the violation of Definitions 3, 4, even if the private data cannot be singled out with this information.

## III. Privacy analysis of projected gradient ascent algorithm

Let us first describe the optimization algorithm used for solving the minimization problem, without discussing the privacy guarantees. To this end, we resort to duality theory [23, Ch. 5], and show how to retrieve the optimal value of the primal from the optimal value of the dual.

For a convex quadratic optimization problem, its dual is also a quadratic optimization problem:

$$\mu^* = \underset{\mu\in\mathbb{R}^m}{\operatorname{argmax}} \; -\frac{1}{2}(A_{\mathcal{C}}^\intercal\mu + c_{\mathcal{A}})^\intercal Q_{\mathcal{C}}^{-1}(A_{\mathcal{C}}^\intercal\mu + c_{\mathcal{A}}) - \mu^\intercal b_{\mathcal{A}}$$
$$s.t.\ \mu \succeq 0. \tag{2}$$

The dual objective function is denoted by $g(\mu)$ and has the gradient equal to $\nabla g(\mu) = -A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}(A_{\mathcal{C}}^\intercal\mu + c_{\mathcal{A}}) - b_{\mathcal{A}}$. Under constraint qualifications, e.g., Slater's condition, strong duality between the primal and dual holds, which means the optimal objective in the primal problem (1) is equal to the objective in the dual problem (2). Moreover, the optimality conditions (Karuhn-Kush-Tucker) hold and are the following:

$$Q_{\mathcal{C}}x^* + A_{\mathcal{C}}^\intercal\mu^* + c_{\mathcal{A}} = 0 \tag{3}$$
$$A_{\mathcal{C}}x^* - b_{\mathcal{A}} \preceq 0, \quad \mu^* \succeq 0 \tag{4}$$
$$\mu_i^*(a_i^\intercal x^* - b_i) = 0,\ i = 1,\dots,m. \tag{5}$$

For strictly convex problems, i.e., $Q_{\mathcal{C}} \in \mathbb{S}_{++}^n$, the optimal solution of the primal problem can be obtained as $x^* = -Q_{\mathcal{C}}^{-1}(A_{\mathcal{C}}^\intercal\mu^* + c_{\mathcal{A}})$.

### A. Projected gradient ascent

A reliable algorithm for computing the optimum in problem (2) – that is also compatible with partially homomorphic encryption, as we will discuss in Section IV-A – is the projected gradient ascent method. The projected gradient ascent is composed by iterations of the following type:

$$\mu_{k+1} = \max\{\mathbf{0}, \mu_k + \eta\nabla g(\mu_k)\}, \tag{6}$$

where $\eta > 0$ is the step size and $\mu_{k+1}$ is the projected value of $\mu_k + \eta\nabla g(\mu_k)$ over the non-negative orthant. For full rank of $A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}A_{\mathcal{C}}^\intercal$, the dual problem is strictly convex and the algorithm converges with a linear rate [24] for a fixed step size $\eta = \frac{1}{L}$, where $L = \lambda_{max}(A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}A_{\mathcal{C}}^\intercal)$. For non-strictly convex dual function, the gradient ascent algorithm converges in sublinear time.

### B. Information leakage from the dual

We want to produce a protocol, such that under the knowledge of the system's architecture and the algorithm used, an adversary (that either controls one party or a coalition of the parties in the protocol) is not capable of inferring private data, i.e., more than they already know. Therefore, we must discuss the information leaked by an optimization algorithm employed for solving Problem (1) and, at the same time, Problem (2). In what follows, we underline the challenges faced by a secure protocol, in the sense of Definitions 3, 4.

Due to the fact that probabilistic encryptions do not preserve the order of the inputs (Section IV-A), the comparison between ciphertexts cannot be executed by a party that does not have access to the decryption key and requires a two-party protocol. Therefore, (6) needs to be performed between the cloud and the target. We will further argue that privacy is lost if the sign of the dual variable is revealed by the protocol to any party (i.e. the cloud or target), due to the complementary slackness conditions (5).

1) Consider a fictitious protocol in which the cloud is given access to $sgn(\mu_{k+1})$, meaning it knows if it is zero or strictly positive. Denote by $\mathcal{I}_+ := \{i|\mu_i^* > 0\}$ and $\mathcal{I}_0 := \{i|\mu_i^* = 0\}$ the sets at the optimum. Therefore, the cloud knows from the Karuhn-Kush-Tucker conditions that:
$$a_i^\intercal x^* = b_i,\ i \in \mathcal{I}_+, \quad q_j^\intercal x^* + \sum_{i\in\mathcal{I}_+} a_{ji}\mu_i^* = -c_j, \tag{7}$$
where $j = 1,\dots,n,$, $a_{ji}$ are the elements of matrix $A_{\mathcal{C}}$ and $q_j$ are the rows of matrix $Q_{\mathcal{C}}$, both known to the cloud, and $b_i, c_i$ are the elements of $b_{\mathcal{A}}, c_{\mathcal{A}}$. If the cloud forms a coalition with a number of agents, subspaces in which $x^*$ lies can be computed.
2) Consider now a protocol in which the target node is given access to $sgn(\mu_{k+1})$. Then, the target will also know that equation (7) holds. If the matrices $A_{\mathcal{C}}$ and $Q_{\mathcal{C}}$ are public information, the target node can immediately compute the values of $b_i$ for $i \in \mathcal{I}_+$ and infer information about $c_{\mathcal{A}}$, which leads to loss of privacy.

By the above analysis, it is clear that we need a protocol that does not reveal the result of the projection of the dual variable to any of the parties. We will present such a protocol in Section IV-B.

### C. Side-information

In the above section, we covered the information that should be hidden by a protocol. However, information such as the architecture of the system, the type of problem solved and the steps of the protocol are public information and cannot be hidden. Therefore, every party knows the feasibility conditions for both the primal and the dual problem, i.e. (4). If there exists information about the probability distribution of the dual variable – e.g., it is more likely that the majority of the constraints are satisfied with equality rather than with strict inequality or vice-versa – a secure algorithm with respect to Definitions 3, 4 cannot hide this information, since it is inherent to the problem, but it does not amplify the information leakage.

Furthermore, in the scenario where the target node colludes with a number of agents, by verifying if the corresponding constraints are inactive at the optimum, the coalition can determine that the associated dual variable is zero. This is side-information arising from duality theory, and no protocol can hide this. Definitions 3, 4 and the subsequent

discussion capture the notion of security that can be offered by the execution of a protocol under side-information.

### D. Comparison to [16]

In the protocol proposed in [16], the target node obtains the sign of the dual variable from the protocol specification. Then, the information leakage discussed in Section III-B takes place and, since the matrices $A_{\mathcal{C}}$ and $Q_{\mathcal{C}}$ are public information, an adversary can successfully determine the private data of the other participants in the protocol. Nevertheless, the protocol satisfies Definition 4 because the sensitive information revealed is prescribed as an output to the target. However, we will show that the optimal solution can be achieved without revealing this crucial information. It is worth noting that our secure protocol will be more computationally intensive than the protocol presented in [16], due to the extra steps that guarantee privacy. In the next section, we will describe a secure comparison and update protocols that correctly compute the projection on the positive orthant without revealing the result to either of the parties.

## IV. Privacy-preserving quadratic optimization

### A. Partially homomorphic encryption scheme

Partially homomorphic encryption schemes can support either multiplications between encrypted data, such as El Gamal [19], unpadded RSA [25] or additions between encrypted data, such as Paillier [17], Goldwasser-Micali [26], DGK [27]. We will focus on the latter, more specifically, the Paillier cryptosystem. For a plaintext message $a$, let $E(a)$ define a generic encryption primitive. Then, the property of additively homomorphic schemes is that there exists an addition operator $\oplus$ defined on the space of ciphertexts such that $E(a) \oplus E(b) \in E(a+b)$, for any plaintexts $a, b$ supported by the schemes. Here, we use set membership instead of equality because the encryption of a message is not unique in some cryptosystems. It is immediate to see that if a scheme supports addition between encrypted messages, it will also support subtraction, by adding a negative number, and multiplication between a plaintext and an encrypted message, obtained by adding the encrypted messages for the corresponding number of times. Hence, such a cryptosystem befits a gradient method for the quadratic optimization problem (1), where the equations are linear in the private data.

We will now describe the encryption scheme used in the following protocols and the specific additive homomorphisms. The **Paillier cryptosystem** [17] is an additively homomorphic encryption scheme. The public key is $pk_{Paillier} = (N, g)_{Paillier}$ and the secret key is $sk_{Paillier} = (\gamma, \delta)_{Paillier}$, where $N$ is the product of two large prime numbers $p, q$, and $g$ is a generator of order $N$ of the group $\mathbb{Z}_N$. The secret key is $\gamma = lcm(p-1, q-1)$ and $\delta = ((g^{\gamma} \mod N^2 - 1)/N)^{-1} \mod N$. A simpler variant to generate the pair of keys when $p$ and $q$ have the same number of bits is to choose $g = N + 1$, $\gamma = \phi(N) = (p-1)(q-1)$, where $\phi$ is Euler's totient function and $\delta = \phi(N)^{-1} \mod N$.

For a plaintext $a \in \mathbb{Z}_N$, the Paillier encryption is $[[a]] = g^a r^N \mod N^2$, where $r$ is a random integer in $\mathbb{Z}_N$, which makes Paillier a probabilistic encryption scheme. Therefore, the ciphertexts do not preserve the order relation between the plaintexts. The decryption follows as $a = ((([[a]]^{\gamma} \mod N^2 - 1)/N)\delta \mod N$ using the fact that $(1+N)^a = 1 + Na \mod N^2$. In this cryptosystem, in order to obtain addition between plaintexts, the operation between ciphertexts is modular multiplication, since $[[a]] \cdot [[b]] = g^a r^N \mod N^2 \cdot g^b r'^N \mod N^2 = g^{a+b}(rr')^N \mod N^2 \in E(a+b)$. Similarly, we use the modular inverse to achieve the negation because $[[a]]^{-1} = g^{-a}(r^{-1})^N \mod N^2 \in E(-a)$.

Therefore, the operations are denoted as follows: addition between two encrypted values as $[[a]] \cdot [[b]]$, difference between two encrypted values as $[[a]] \cdot [[b]]^{-1}$ and multiplication between a plaintext and an encrypted value as $[[a]]^b$. We will slightly abuse the last notation to denote additions and multiplication by vectors and matrices. Moreover, for everything that follows, we denote by $[[\cdot]]$ the encryption with the target node's public key $pk_{\mathcal{T}}$.

The optimization problem (1) is defined on real variables, whereas the encryption scheme is defined on integers. We consider a fixed-point representation of the values in order to deal with aspect.

### B. Secure comparison blocks

Consider a two-party computation problem under an encryption scheme that does not support comparison between encrypted data. A number of secure comparison protocols on private inputs owned by two parties have been proposed in the literature [27], [28], [29], [30] etc., with a survey of the state of the art given in [31]. Most of the comparison protocols have linear complexity in the size of the inputs, since the comparison is done bitwise. Out of these, [27] with the correction in [32] remains one of the most computationally efficient protocols, and [29] is the most efficient for large-scale protocols. [30] proposes a comparison that is sublinear in the number of invocation of the cryptographic primitive, but has greater communication complexity and is only competitive for large inputs, due to the constants involved. Depending on the specific problem, some variants might be better than others. Since this comparison protocol is used as a block, it can be easily replaced once more efficient protocol are proposed.

We will demonstrate how the comparison protocol works via the DGK protocol. Damgård, Geisler and Krøigaard describe a protocol in [27], [32] for secure comparison between two private inputs of different parties. To this end, they also propose the DGK additively homomorphic encryption scheme with the property that it is efficient to determine if the value zero is encrypted, which is useful for comparisons. An extension of this protocol to the case where none of the parties knows the two numbers that have to be compared, which is of interest to us, and some improvements in terms of computation and efficiency were proposed in [33].

We will now outline a general comparison protocol that will be used in our optimization protocol. Consider two semi-honest parties A and B, which hold the place of the cloud and target node in our setup. Let A have two encrypted values under Paillier's scheme $[[a]]$ and $[[b]]$, and B have the decryption key. At the end of the protocol, the party B will

have the result of the comparison in the form of one bit $t$ such that $(t = 1) \Leftrightarrow (a \leq b)$. Let $l$ denote the number of bits of the unencrypted inputs $a, b$. Protocol 1 is based on the fact that the most significant bit of $(b - a + 2^l)$ is the bit that indicates if $(a \leq b)$. The random numbers used for blinding the values in Protocols 1, 3 are sampled uniformly from $(0, 2^{\lambda+l}) \cap \mathbb{Z}_N$, where $\lambda$ is the security parameter of length at least 80 bits, chosen such that brute-forcing the solution is intractable. In order to guarantee correctness, no overflow must take place, so we must impose $log_2 N > l + \lambda + 1$.

Protocol 1 calls the DGK protocol. Due to space constraints, we will only describe the idea behind the protocol for two-party secure comparison with private inputs. In the DGK protocol, there are two parties A and B, each with their own private input $r$, respectively $z$. Moreover, the parties have access to the value $l$, such that they compute $A : \alpha \leftarrow r \mod 2^l$ and $B : \beta \leftarrow z \mod 2^l$. Using the binary representations of $\alpha$ and $\beta$, the two parties exchange $l$ blinded values such that A obtains the encrypted bit $[[t']]$ that satisfies $(t' = 1) \Leftrightarrow (\beta < \alpha)$.

PROTOCOL 1: Protocol for secure two-party comparison with two encrypted inputs using DGK [27], [33]

**Require:** A: $[[a]], [[b]]$; B: $sk_{Paillier}, sk_{DGK}$
**Ensure:** B: encrypted bit $t$: $(t = 1) \Leftrightarrow (a \leq b)$
1: A: choose random number $r$
2: A: $[[z]] \leftarrow [[b]] \cdot [[a]]^{-1} \cdot [[2^l + r]] \mod N^2$, send $[[z]]$ to B
▷ $z \leftarrow b - a + 2^l + r$
3: B: decrypts $[[z]]$
4: A: $\alpha \leftarrow r \mod 2^l$
5: B: $\beta \leftarrow z \mod 2^l$
6: A, B: privately compute an encrypted bit $[[t']]$ with a comparison protocol, e.g, DGK, such that $(t' = 1) \Leftrightarrow (\beta < \alpha)$
7: B: compute $[[z \div 2^l]]$ and send it to A
8: A: $[[t]] \leftarrow [[z \div 2^l]] \cdot [[r \div 2^l]]^{-1} \cdot [[t']]^{-1} \mod N^2$
9: A: send $[[t]]$ to B

*Proposition 1 ([27], [32], [33]):* Protocol 1 is secure in the semi-honest model.

*Remark 2:* The protocol for secure comparison between two private inputs of two parties presented above is also secure when the two inputs are equal, i.e. $a = b$. For the problem presented in this paper, this detail is crucial, since it is the case when $\mu_k + \eta \nabla g(\mu_k) = 0$. Being able to distinguish equality between inputs from inequality would reveal the value of $\mu_{k+1}$.

### C. Secure value update

As we discussed in Section III, we want to keep the result of the comparison of the updated iterate with zero unknown to both of the parties. Notice that Algorithm 1 reveals the result of the comparison between $a$ and $b$ to party B, which in our setup is the target node $\mathcal{T}$. However, if we introduce an additional step where $\mathcal{C}$ randomizes the order of the two values that it wants to compare, then $\mathcal{T}$ does not learn any information by knowing the result of the comparison.

PROTOCOL 2: Randomization step to prevent the target node of making use of the result of Algorithm 1

**Require:** $\bar{\mu}_k := \mu_k + \eta \nabla g(\mu_k)$, $\mathcal{C}$: $[[\bar{\mu}_k]], [[0]]$
**Ensure:** $\mathcal{C}$: $[[a]], [[b]]$
1: $\mathcal{C}$: choose a random bit $r$
2: **if** $r = 0$ **then**
3: $\quad \mathcal{C} : [[a]] \leftarrow [[\bar{\mu}_k]], [[b]] \leftarrow [[0]]$
4: **else**
5: $\quad \mathcal{C} : [[b]] \leftarrow [[\bar{\mu}_k]], [[a]] \leftarrow [[0]]$
6: **end if**

Moreover, we need to ensure that when the cloud $\mathcal{C}$ updates the value of the dual variable at iteration $k + 1$ in equation (6), it does not know the value it updates it with. The solution we propose is that the cloud must blind the values of $[[a]]$ and $[[b]]$ and send them to the target node in this order, where the target will select the value accordingly to the comparison result and then send it back to the cloud. However, there are two important issues that have to be addressed in order for the update to not leak information about the sign of $\bar{\mu}_k$: the blinding should be additive and effectuated with different random values, and the ciphertexts should be refreshed. The reasons are the following: if the blinding is multiplicative, by decrypting the product, the target knows which one of the values is zero. Moreover, if the two values are additively blinded with the same random value, the target can subtract $a + r - (b + r)$ and obtain either $\bar{\mu}_k$ or $-\bar{\mu}_k$, which reveals at least if the value is zero. Re-randomization of the encryptions is necessary, such that the cloud cannot simply compare $[[a]]$ and $[[b]]$ with the received value. This can be done by adding an encryption of zero or by decryption followed by encryption, which also avoids the propagation of the errors due to the fixed-point representation. We propose Protocol 3 as the solution to the update problem:

PROTOCOL 3: Secure update of the dual variable

**Require:** $\mathcal{C}$: $[[a]], [[b]]$; $\mathcal{T}$: $t_k$ such that $(t_k = 1) \Leftrightarrow (a \leq b)$
**Ensure:** $\mathcal{C}$: $[[\mu_{k+1}]]$
1: $\mathcal{C}$: choose two random numbers $r_k$, $s_k$
2: $\mathcal{C}$: $[[\bar{a}]] \leftarrow [[a]] \cdot [[r_k]], [[\bar{b}]] \leftarrow [[b]] \cdot [[s_k]]$
3: $\mathcal{C}$: send $[[\bar{a}]]$ and $[[\bar{b}]]$ to $\mathcal{T}$
4: **if** $t = 0$ **then** $\mathcal{T}$: $[[v_k]] \leftarrow \widetilde{[[\bar{a}]]}$
5: **else** $\mathcal{T}$: $[[v_k]] \leftarrow [[\bar{b}]]$
6: **end if** ▷ Refresh the ciphertext
7: $\mathcal{T}$: send $[[v_k]]$ and $[[t_k]]$ to $\mathcal{C}$
8: $\mathcal{C}$: $[[\mu_{k+1}]] \leftarrow [[v_k]] \cdot (g^{-1}[[t_k]])^{r_k} \cdot [[t_k]]^{-s_k}$
▷ $\mu_{k+1} \leftarrow v_k + r_k(t_k - 1) - s_k t_k$

Here, $g$ is the generator of the multiplicative group in Paillier's encryption. If $a \leq b$, then $t_k = 1$ and we obtain $\mu_{k+1} = \bar{b}_k - s_k = b_k$, and otherwise, $t_k = 0$ and we obtain $\mu_{k+1} = \bar{a}_k - r_k = a_k$.

Having defined these protocols, we can now build a protocol that represents an iteration in the dual projected gradient ascent method. Throughout this paper, by comparison we mean element-wise comparison, since $\mu$ is a vector and we project it on $\mathbb{R}^m_+$.

PROTOCOL 4: Secure iteration of the dual projected gradient ascent method

**Require:** $\mathcal{C}$: $A_\mathcal{C} \in \mathbb{R}^{m \times n}, Q_\mathcal{C} \in \mathbb{S}^n_{++}, [[b_\mathcal{A}]], [[c_\mathcal{A}]], \eta > 0, [[\mu_k]]$; $\mathcal{T}$: $sk_\mathcal{T}$
**Ensure:** $\mathcal{C}$: $[[\mu_{k+1}]]$
1: $\mathcal{C}$: $[[\nabla g(\mu_k)]] \leftarrow [[\mu_k]]^{-A_\mathcal{C} Q_\mathcal{C}^{-1} A_\mathcal{C}^\intercal} \cdot [[c_\mathcal{A}]]^{-A_\mathcal{C} Q_\mathcal{C}^{-1}} \cdot [[b_\mathcal{A}]]^{-1}$
▷ Compute the encrypted gradient

2: $\mathcal{C}$: $[[\bar{\mu}_k]] \leftarrow [[\mu_k]] \cdot [[\nabla g(\mu_k)]]^{\eta}$ ▷ Update the value in the ascent direction
3: $\mathcal{C}$ execute Protocol 2: $\mathcal{C}$ gets $[[a]], [[b]]$ ▷ Randomly assign $[[a]], [[b]]$ with values of $[[\bar{\mu}_k]], [[0]]$
4: $\mathcal{C}, \mathcal{T}$ execute Protocol 1 element-wise: $\mathcal{T}$ gets $t_k$ ▷ Secure comparison protocol, where party A $\equiv \mathcal{C}$ and party B $\equiv \mathcal{T}$
5: $\mathcal{C}, \mathcal{T}$ execute Protocol 3: $\mathcal{C}$ obtains $[[\mu_{k+1}]]$ ▷ Secure update protocol that ensures $\mu_{k+1} = \max\{\bar{\mu}_k, 0\}$

The proof of security in the semi-honest model is omitted, but follows similar steps as in [34].

*Proposition 2 ([34] Protocol 1 -* argmax*):* Protocol 4 is secure in the semi-honest model.

### D. Protocol for solving strictly-convex quadratic problem

Using the building blocks described above, we can finally assemble the algorithm that securely solves a quadratic optimization problem with private data and sends the optimal solution to a target node. The public key $pk_{\mathcal{T}}$ is known by all the parties, hence we omit it from the inputs.

PROTOCOL 5: Privacy preserving algorithm for solving strictly-convex quadratic optimization problems

**Require:** $\mathcal{A}_{i=1,\ldots,p}$: $b_{\mathcal{A}} = \{b_j\}_{j=1,\ldots,m}, c_{\mathcal{A}} = \{c_k\}_{k=1,\ldots,n}$; $\mathcal{C}$: $A_{\mathcal{C}} \in \mathbb{R}^{m \times n}, Q_{\mathcal{C}} \in \mathbb{S}^n_{++}, K$; $\mathcal{T}$: $sk_{\mathcal{T}}, K$
**Ensure:** $\mathcal{T}$: $x^*$
**for** i=1,...,p **do**
  $\mathcal{A}_i$ : encrypt the private information $msg_i \leftarrow ([[b_i]], [[c_i]])$
  $\mathcal{A}_i$ : send the encrypted messages to $\mathcal{C}$
**end for**
$\mathcal{C}$: Construct the vectors $[[b_{\mathcal{A}}]]$ and $[[c_{\mathcal{A}}]]$ from the messages
$\mathcal{C}$: $\eta \leftarrow 1/\lambda_{max}(A_{\mathcal{C}} Q_{\mathcal{C}}^{-1} A_{\mathcal{C}}^{\intercal})$
$\mathcal{C}$: Choose a random initial value $\mu_0$ for the dual variable and encrypt it: $[[\mu_0]]$
**for** each $k = 0, \ldots, K$ **do**
  $\mathcal{C}, \mathcal{T}$ execute Protocol 4: $\mathcal{C}$ gets $[[\mu_{k+1}]]$ ▷ $\mathcal{C}, \mathcal{T}$ securely effectuate an iteration of the dual projected gradient ascent
**end for**
$\mathcal{C}$: $[[x_K]] \leftarrow [[\mu_K]]^{-Q_{\mathcal{C}}^{-1} A_{\mathcal{C}}^{\intercal}} \cdot [[c_{\mathcal{A}}]]^{-Q_{\mathcal{C}}^{-1}}$ and send it to $\mathcal{T}$ ▷ Compute the primal optimum from the optimal dual solution
$\mathcal{T}$: Decrypt $[[x_K]]$ and output $x^*$

### E. Computational complexity of the protocol

The efficiency of the algorithm is measured in the number modular operations, encryptions and decryptions (which are composed of modular multiplications) and exchanges, which are relevant when the communication is slow. These are objective measurements of efficiency, while the duration of each execution highly depends on the CPU power of the machines that run the protocol. In the setup we considered, the agents are low-power machines, hence, they are only required to effectuate one encryption and one communication round, but the cloud and the target node are servers, with high computational capabilities. In this paper, we outlined the logical flow of the protocols, but there are several optimizations possible in order to reduce the computational and communication complexity. For example, in order to save online computations, the coins used for encryptions and for additive blinding can be precomputed. Similarly, the dot products, element-wise comparisons, additions, encryptions etc. can be executed in parallel, which saves an order of $m$ for the variables in $\mathbb{R}^m$ and $n$ for the variables in $\mathbb{R}^n$.

Let $\sigma$ be the bit-length of the modulus $N$. A Paillier encryption of an $l$-bit plaintext takes $\mathcal{O}(l)$ multiplications modulo $N^2$, and a decryption takes $\mathcal{O}(\sigma)$ multiplications modulo $N$. A multiplication of an encrypted value with a plaintext of $l$ bits is achieved as a modular exponentiation, which takes $\mathcal{O}(l)$ multiplications modulo $N^2$. The comparison protocol takes around $O(l^2)$ modular multiplications with modulus $N$ and $O(l)$ messages of size $\sigma$, with variations depending on the subroutine of comparing private inputs. Roughly, Protocol 5 takes $\mathcal{O}(Kml^2)$ modular multiplications in $\mathbb{Z}_N$, $\mathcal{O}(K(lm^2+lmn))$ modular multiplications in $\mathbb{Z}_{N^2}$, $\mathcal{O}(Km)$ $2\sigma$-bit messages and $\mathcal{O}(Klm)$ $\sigma$-bit messages exchanged.

### F. Fast Gradient Method

The larger the number of iterations required for an optimization problem, the more delay and complexity are incurred due to the cryptography applied at each iteration. Therefore, apart from the optimization at the level of computations, we can also achieve a speedup of the above algorithm through a smaller number of iterations, in another version of the gradient method. The fast gradient method introduced by Nesterov (see e.g. [24]) converges in superlinear time for a strongly convex objective function. It is straightforward to adapt Protocol 5 such that we implement the fast gradient method for the projection on the non-negative orthant.

## V. SECURITY OF THE PROPOSED PROTOCOL

We will now discuss the security of Protocol 5. Proving security in the semi-honest model involves the concepts of semantic security of an encryption scheme. Under the assumptions of decisional composite residuosity [17] and hardness of factoring, the partially homomorphic encryption schemes Paillier and DGK are semantically secure and have indistinguishable encryptions, which, in essence, means that we cannot distinguish an encryption of a value $[[x]]$ from an encryption of another value $[[y]]$. Furthermore, we need to introduce the concepts of ideal, real and hybrid models of executing protocols [21], [35] in order to construct the ppt algorithm that simulates the execution of the protocol, as in Definitions 3 and 4. To simplify the complexity of the proof, we use the theorem of sequential modular composition [36], [21], [35], which states that if a protocol is secure in the semi-honest model, then it remains secure under ordered composition with other secure protocols.

*Definition 5:* (Semantic Security [22, Ch. 5]) An encryption scheme is **semantically secure** if for every ppt algorithm, A, there exists a ppt algorithm $A'$ such that for every two polynomially bounded functions $f, h : \{0,1\}^* \rightarrow \{0,1\}^*$ and for any probability ensemble $\{X_n\}_{n \in \mathbb{N}}$, $|X_n| \leq poly(n)$, for any positive polynomial $p$ and sufficiently large $n$, where $E(\cdot)$ is the encryption primitive:

$$\Pr[A(E(X_n), h(X_n)) = f(X_n)] < \Pr[A'(h(X_n)) = f(X_n)] + 1/p(n),$$

*Definition 6:* (f-hybrid model [35]) In a hybrid model, two parties interact with each other (as in the real model) and run a protocol $\Pi$ that uses calls to a trusted third-party (as in the ideal model) that computes a functionality $f$.

Let A be a non-uniform ppt machine and let $I$ denote a coalition of corrupted parties. Then, the **f-hybrid execution of Π** on inputs $\bar{x}$ and auxiliary input $z$ to A, denoted $\text{HYBRID}^{f}_{\Pi,A(z),I}(\bar{x})$ is defined as the output vector of the honest parties and the adversary A from the hybrid execution of Π, with a trusted party computing $f$.

*Theorem 1:* (Sequential modular composition [35]) Let $f_1, \ldots, f_s$ be two-party ppt functionalities and let $\rho_1, \ldots, \rho_s$ be protocols such that each $\rho_i$ securely computes $f_i$ in the presence of semi-honest adversaries. Let $g$ be a two-party ppt functionality, and let Π be a protocol that securely computes $g$ in the $f_1, \ldots, f_s$-hybrid model in the presence of semi-honest adversaries. Then, $\Pi^{\rho_1,\ldots,\rho_s}$ securely computes $g$ in the presence of semi-honest adversaries.

The real protocol $\Pi^{\rho_1,\ldots,\rho_s}$ is defined as follows: all standard messages of Π are unchanged; when a party is instructed to send an ideal message $\alpha$ to the trusted party to compute $f_i$, it begins with a real execution of $\rho_i$ with input $\alpha$. When this execution of $\rho_i$ concludes with an output $\beta$, the party continues the execution of Π as if $\beta$ was the output received by the trusted party, which is the same as if the party were running in the $f_i$-hybrid model.

We can now state and prove the main theorem. Due to space constraints, we only provide a sketch of the proof.

*Theorem 2:* Protocol 5 is secure in the semi-honest model.

*Sketch of Proof*: Consider an iteration of Protocol 5. Firstly, the Paillier cryptosystem is semantically secure, therefore, two ciphertexts are computationally indistinguishable by a party that does not have access to the secret key. Secondly, every exchange is additively blinded using a different random number, uniformly sampled from a large enough range (at least $\lambda$ bits more over the size of the value to be blinded, where $\lambda$ is the security parameter), which means that the blinded message is computationally indistinguishable from a random number sampled from the same distribution. Thirdly, the ciphertexts are refreshed after each exchange, so a party that does not have access to the decryption key cannot infer information about the encrypted values by simply comparing the ciphertexts. Then, none of the parties can infer the magnitude or the sign of the values they are receiving, which proves security for this problem. Formally, using Definition 3, the view of a simulator that generates random values for its transcript and the view obtained by the parties running an iteration of the protocol on the real inputs are computationally indistinguishable. Due to the same reasons as above, storing the exchanged messages does not give any new information on the private data of the honest parties. Hence, by using a hybrid argument and calling the ideal functionality of an iteration $K$ times, we prove the security of the protocol in the semi-honest model invoking the sequential composition theorem. ■

We now show that any coalition consisting of a number of agents and the cloud or of a number of agents and the target node gains no information of the private data of the honest parties. A coalition between the cloud and the target node is impossible, since such a coalition would be omniscient. Similarly, only coalitions of up to $\bar{p} \leq p-1$ agents are possible, since a coalition consisting of all the agents has access to all the private inputs.

*Theorem 3:* i) Protocol 5 is secure in the semi-honest model under any coalition between the cloud $\mathcal{C}$ and a set of agents $\mathcal{A}_{i=1,\ldots,\bar{p}}$.
ii) Protocol 5 is secure in the semi-honest model under any coalition between the target $\mathcal{T}$ and a set of agents $\mathcal{A}_{i=1,\ldots,\bar{p}}$.

*Sketch of Proof*: For each coalition of parties, the proof shows there exists a simulator, as in Definition 4, that has the same view as the coalition during the execution of the protocol. Apart from the initial step of sending their encrypted inputs to the cloud, the agents do not contribute to Protocol 5. Therefore, a coalition between a set of agents $\mathcal{A}_{i=1,\ldots,\bar{p}}$ and either the cloud $\mathcal{C}$ or the target node $\mathcal{T}$ can be modeled through the auxiliary information in the two-party case. Specifically, introduce in the auxiliary input the following: $(\{b_i\}_{i=1,\ldots,\bar{p}}, \{c_i\}_{i=1,\ldots,\bar{p}})$. Then, this proof follows from the proof of Theorem 2, by the fact that every value the target node receives is additively blinded by an appropriate selected noise and every value the cloud receives in encrypted by a semantically secure cryptosystem. ■

*Remark 3:* The security of Protocol 5 holds indifferent of the privateness of the matrices $A_{\mathcal{C}}$ and $Q_{\mathcal{C}}$ and the step $\eta$, i.e., if they were public information instead of private information of the cloud $\mathcal{C}$. This is due to the fact that every value the target node receives is additively blinded by an appropriate selected noise and every value the cloud receives in encrypted by a semantically secure cryptosystem, and does not depend on whether the elements in $A_{\mathcal{C}}$ and $Q_{\mathcal{C}}$ are known or not.

## VI. Equality constraints

Consider now problem (1) with additional linear equality constraints $H_{\mathcal{C}}x = d_{\mathcal{A}}$, where $H_{\mathcal{C}} \in \mathbb{R}^{q\times n}, d_{\mathcal{A}} \in \mathbb{R}^{q}$. If $H_{\mathcal{C}}$ is public, then $d_{\mathcal{A}}$ should also be public, since the target can immediately compute it from $H_{\mathcal{C}}x^* = d_{\mathcal{A}}$.

The dual of the new problem is the following:

$$\begin{aligned}\max_{\substack{\mu\in\mathbb{R}^m\\ \nu\in\mathbb{R}^q}} \; & -\frac{1}{2}(c_{\mathcal{A}} + A_{\mathcal{C}}^{\intercal}\mu + H_{\mathcal{C}}^{\intercal}\nu)^{\intercal}Q_{\mathcal{C}}^{-1}(c_{\mathcal{A}} + A_{\mathcal{C}}^{\intercal}\mu + H_{\mathcal{C}}^{\intercal}\nu) - \\ & - b_{\mathcal{A}}^{\intercal}\mu - d_{\mathcal{A}}^{\intercal}\nu \\ s.t.\; & \mu \succeq 0\end{aligned}$$

We want to modify the gradient ascent algorithm to solve this problem. The optimal value of the primal problem becomes $x^* = -Q_{\mathcal{C}}^{-1}(A_{\mathcal{C}}^{\intercal}\mu^* + H_{\mathcal{C}}^{\intercal}\nu^* + c_{\mathcal{A}})$. Since the variable $\nu$ is not constrained in the dual problem, there is no need for projection in the dual optimization algorithm. Therefore, the iterations in the modified protocol will be the following:

$$\begin{aligned}[[\nabla_{\mu} g(\mu_k,\nu_k)]] &\leftarrow [[\mu_k]]^{-A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}A_{\mathcal{C}}^{\intercal}} \cdot [[\nu_k]]^{-A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}H_{\mathcal{C}}^{\intercal}} \cdot \\ &\quad \cdot [[c_{\mathcal{A}}]]^{-A_{\mathcal{C}}Q_{\mathcal{C}}^{-1}} \cdot [[b_{\mathcal{A}}]]^{-1} \\ [[\nabla_{\nu} g(\mu_k,\nu_k)]] &\leftarrow [[\nu_k]]^{-H_{\mathcal{C}}Q_{\mathcal{C}}^{-1}H_{\mathcal{C}}^{\intercal}} \cdot [[\mu_k]]^{-H_{\mathcal{C}}Q_{\mathcal{C}}^{-1}A_{\mathcal{C}}^{\intercal}} \cdot \\ &\quad \cdot [[c_{\mathcal{A}}]]^{-H_{\mathcal{C}}Q_{\mathcal{C}}^{-1}} \cdot [[d_{\mathcal{A}}]]^{-1} \\ [[\mu_{k+1}]] &\leftarrow \max\{[[0]], [[\mu_k]] \cdot [[\nabla_{\mu} g(\mu_k,\nu_k)]]^{\eta}\} \\ [[\nu_{k+1}]] &\leftarrow [[\nu_k]] \cdot [[\nabla_{\nu} g(\mu_k,\nu_k)]]^{\eta}\end{aligned}$$

The convergence of this algorithm is the same as the convergence of the previously discussed projected gradient algorithm for the step-size $\eta = 1/\lambda_{max}(\tilde{Q})$, where

$\tilde{Q} = \begin{bmatrix} A_C Q_C^{-1} A_C^\intercal & A_C Q_C^{-1} H_C^\intercal \\ H_C Q_C^{-1} A_C^\intercal & H_C Q_C^{-1} H_C^\intercal \end{bmatrix}$. The proof of privacy in the semi-honest model follows from Theorem 2.

## VII. Conclusions

This work explored privacy-preserving convex quadratic problems with strictly convex objective function and linear constraints. We considered a multi-party framework, where a set of agents outsource their encrypted data to an untrusted cloud, which solves a quadratic problem over their data. The cloud communicates with another untrusted entity, called the target node, to which it has to send the solution of the convex optimization problem. We discussed the security definitions for the multi-party computation model and showed the information due to duality theory revealed by optimization algorithms. In order to overcome this issue, we assembled a protocol that implements the dual gradient ascent method and presented secure comparison and secure update protocols. Under an additively homomorphic encryption scheme, the protocol preserves the overall privacy of a quadratic optimization problem with sensitive data. Our proposed protocol also allows linear equality constraints and extends to accelerated gradient methods.

In future work, we will also consider secure protocols for distributed private quadratic cost, i.e., the cloud does not have access to the $Q$ matrix, which implies nonlinear operations between encrypted data. Furthermore, we want to generalize the class of optimization problems that we can securely solve. We will also carry out implementation and applications on real problems, to evaluate the feasibility and efficiency of the proposed algorithms in practice.

## References


[1] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *52nd Design Automation Conference*. IEEE, 2015, pp. 1–6.

[2] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, 2013.

[3] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A survey on security issues and solutions at different layers of cloud computing," *The Journal of Supercomputing*, vol. 63, no. 2, pp. 561–592, 2013.

[4] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *Proceedings of the Conference on Computer and Communications Security*. ACM, 2012, pp. 784–796.

[5] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2014, pp. 533–556.

[6] R. Cramer, I. B. Damgård *et al.*, *Secure Multiparty Computation*. Cambridge University Press, 2015.

[7] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[8] L. J. Aslett, P. M. Esperança, and C. C. Holmes, "A review of homomorphic encryption and software tools for encrypted statistical machine learning," *arXiv preprint arXiv:1508.06574*, 2015.

[9] T. K. Frederiksen, "The hitchhiker's guide to garbled circuits," Ph.D. dissertation, Dept. of Computer Science, Aarhus University, 2015.

[10] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2017.

[11] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the International Conference on Distributed Computing and Networking*. ACM, 2015, pp. 1–10.

[12] S. Han, W. K. Ng, L. Wan, and V. C. Lee, "Privacy-preserving gradient-descent methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 6, pp. 884–899, 2010.

[13] S. Wu, T. Teruya, J. Kawamoto, J. Sakuma, and H. Kikuchi, "Privacy-preservation for stochastic gradient descent application to secure logistic regression," in *27th Annual Conference of the Japanese Society for Artificial Intelligence*, 2013.

[14] T. Toft, "Solving linear programs using multiparty computation," *Financial Cryptography and Data Security*, pp. 90–107, 2009.

[15] O. Catrina and S. De Hoogh, "Secure multiparty linear programming using fixed-point arithmetic," in *European Symposium on Research in Computer Security*. Springer, 2010, pp. 134–150.

[16] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *55th Conference on Decision and Control*. IEEE, 2016, pp. 5053–5058.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.

[18] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Department of Computer Science, Stanford University, 2009, crypto.stanford.edu/craig.

[19] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1984, pp. 10–18.

[20] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011, pp. 505–524.

[21] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2003.

[22] ——, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[25] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[26] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the 14th annual ACM symposium on Theory of Computing*. ACM, 1982, pp. 365–377.

[27] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Australasian Conference on Information Security and Privacy*. Springer, 2007, pp. 416–430.

[28] J. Garay, B. Schoenmakers, and J. Villegas, "Practical and secure solutions for integer comparison," in *International Workshop on Public Key Cryptography*. Springer, 2007, pp. 330–342.

[29] V. Kolesnikov, A.-R. Sadeghi, and T. Schneider, "Improved garbled circuit building blocks and applications to auctions and computing minima," in *International Conference on Cryptology and Network Security*. Springer, 2009, pp. 1–20.

[30] H. Lipmaa and T. Toft, "Secure equality and greater-than tests with sublinear online complexity," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2013, pp. 645–656.

[31] G. Couteau, "Efficient secure comparison protocols," Cryptology ePrint Archive, Report 2016/544, 2016, http://eprint.iacr.org/2016/544.

[32] I. Damgård, M. Geisler, and M. Krøigaard, "A correction to "Efficient and secure comparison for on-line auctions"," *International Journal of Applied Cryptography*, vol. 1, no. 4, pp. 323–324, 2009.

[33] T. Veugen, "Improving the DGK comparison protocol," in *International Workshop on Information Forensics and Security*. IEEE, 2012, pp. 49–54.

[34] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *NDSS*, 2015.

[35] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, p. 5, 2009.

[36] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of Cryptology*, vol. 13, no. 1, pp. 143–202, 2000.