# Convergence Analysis of Iterated Best Response for a Trusted Computation Game [⋆]

Shaunak D. Bopardikar [a], Alberto Speranzon [b], Cédric Langbort [c]

[a] United Technologies Research Center Inc., 2855 Telegraph Avenue Suite 410, Berkeley, CA 94705 USA

[b] United Technologies Research Center, 411 Silver Lane, East Hartford, CT 06118 USA

[c] University of Illinois Urbana Champaign

**Abstract**

We introduce a novel game of trusted computation in which a sensor equipped with limited computing power leverages a central computer to carry out a specified data processing task on a large dataset collected over time. In normal circumstances, the sensor would be able to stream the data to the central computer, which would then perform the computation and provide the result. We assume, however, that the central computer can be under attack and we propose a strategy where the sensor retains a limited amount of the data to counteract the effect of attacks. We formulate the problem within a game theoretic framework where the sensor needs to decide an optimal fusion strategy using both the non-trusted output from the central computer and locally stored trusted data. The sensor generates an estimate of the true computation that is fused with the value from the third-party computation. We adopt an Iterated Best Response (IBR) scheme for each player to update its action based on the opponent's announced computation. At each iteration, the central computer reveals its output to the sensor, who then computes its best response based on a linear combination of its private local estimate and the untrusted third-party output. We characterize equilibrium conditions along with necessary and sufficient conditions for convergence of the IBR. Numerical results are presented showing that the convergence conditions are relatively tight.

*Key words:* Game theory, Computational Methods, Adversarial Machine Learning, Iterated Best Response.

## 1 Introduction and Related Works

The Internet of Things (IoT) [**add citation**] is the next generation internet where many embedded devices are interconnected and can exchange data. Typical examples of such devices are sensors, actuators, controller, etc. Although such devices are becoming increasingly more advanced and capable, the amount of data they can process is still a small fraction of what they can collect. In this context, it is clear that IoT devices need to leverage intermediate but more capable devices that can store and compute over larger data streams.

In this setting, we study the problem where a sensor (a shorthand for IoT device) exchange data with a larger and more powerful computer in order to carry out a computation on data the sensor has collected over a certain period of time. Under normal circumstances, the sensor would be able to send/stream the data to such "central" computer, which would then execute the data processing and send back the result. However, we assume here that the data stored in the central computer can be manipulated maliciously by an attacker. In this case, the sensor has three options. The first is to retain a small amount of the data and compute the function of interest on such trustworthy data, knowing however that the computation will not be as accurate given the small sample size; or, secondly, take the risk that the attack is only mildly compromising the data on the central computer, so that the result of the computation is close to the true value; or, thirdly, try to exchange partial results iteratively with the central computer and fuse locally trusted computation on small sample with tampered computation on the full dataset.

This paper formalizes this last scenario, which has the other two as limiting cases. In particular, we model this problem of trusted computation as a game between the

[⋆] A preliminary version of this paper appeared at the *2015 American Control Conference* under the title *Trusted Computation with an Adversarial Cloud* [3]. Corresponding author: Shaunak D. Bopardikar. Tel. +1-510-649-9974. This work was funded by United Technologies Research Center.

*Email addresses:* bopardsd@utrc.utc.com (Shaunak D. Bopardikar), sperana@utrc.utc.com (Alberto Speranzon), langbort@illinois.edu (Cédric Langbort).

sensor/central computer and the attacker. We design an analyze a protocol in which each player plays its best response and we then study convergence conditions. Furthermore, in this paper we make a "worst case" assumption, namely that the attacker knows exactly the fusion strategy adopted by the sensor. We plan to relax this in future work.

The approach we are considering in this paper is related to a new emerging field, called *adversarial machine learning*, where two parties, a learner and an attacker, are involved, see [8,2,12]. The learner is using the data to train, for example, a classifier or a regressor, and the attacker is modifying the data so that the learner ends up training the algorithm incorrectly. In this context, the problem is posed as a Bayesian game [12], where the learner minimizes the effect of the attack on the learning algorithm, whereas the attacker maximizes the deviation of such learning algorithm from the correct result and towards a strategically chosen outcome, under the assumption that only a subset of the data can be modified. In [12], for example, the e-mail spam problem is considered, where the learner is set to train a classifier to discriminate between spam and non-spam, while the attacker tries to maximize the chances that a spam is classified as non-spam.

The approach considered in this paper relates also to the procedure of *fictitious play (FP)*. In this procedure, each player tries to learn the probability distribution from which the opponent is drawing its actions [4,11]. A recent body of work in the control literature analyzes convergence of fictitious play for several scenarios [13,14,9]. In particular, [13] presents unified energy-based convergence proofs that work for several special classes of games under FP. In [14], convergence to Nash equilibria is analyzed under the assumption that each player can access the derivatives of the update mechanisms, leading to dynamic FP. A variant of FP, known as Joint Strategy FP, is proposed and the convergence analyzed for several classes of games, especially in high-dimensional spaces, see [9]. More recently, Gaussian cheap talk games, such as [6], have been considered. In this context, a sender (adversary) sends corrupted information to a receiver (sensor) under the assumption that the adversary has full knowledge of the receiver's private information.

### *1.1 Main Contributions*

The contributions of this paper are four-fold. First, we formulate a new problem on trusted computation within a game-theoretic framework and adopt an Iterated Best Response (IBR) [10,7] algorithm to compute final strategies for the sensor and the attacker. More specifically, we consider a protocol such that at each iteration, the attacker reveals its output to the sensor that then computes its best response as a linear combination of its private local estimate and of the untrusted output. The attacker can then, based on the announced policy of the sensor, decide its best response. There is a clear mismatch in the information pattern between attacker and sensor and, in particular, the fact that the attacker cannot access the realization of the private local estimate of the sensor distinguishes this work from the information pattern considered in some other existing works such as [6].

Second, we characterize conditions on the existence of equilibria of the game. These conditions and the equilibria themselves turn out to be functions of all of the problem parameters, viz., the private information belonging to both players.

In order to obtain results from a single player's perspective, a third contribution of this paper is to define two notions of convergence for the IBR algorithm, depending upon whether the algorithm converges for some initial value picked by the attacker, *weak convergence*, or for every initial value, *strong convergence*. We derive necessary conditions for weak convergence and sufficient conditions for strong convergence. If the algorithm converges, then it also tells the sensor how to optimally fuse its private estimate with the output. We identify regimes in which some sufficient conditions are also necessary. Numerical simulations indicate that the conditions are relatively tight.

Fourth and finally, the analysis in this paper allows for a certain level of *mismatch* in the distributions used by the players in computing their respective cost functions. This generalizes the analysis presented in the preliminary conference version [3], which assumed that the attacker knows precisely the mean of the distribution used by the sensor to compute its private estimate. Additionally, in the special scenario considered in [3], the analysis in this paper recovers the results from [3], and in one of the cases, also improves the result.

Given that the proposed framework requires an iterative process between sensor and the central computer, the algorithm presented in this paper is suitable for computation algorithms that are iterative in nature so that partial results can be exchanged between sensor and the central computer. Examples are eigenvalue/eigenvector computation, matrix factorization, iterative optimization methods, etc.

The connection of this work to control theory lies in the fact that convergence analysis of the IBR essentially leads to a closed loop dynamical system. This aspect is similar in flavor to the set-ups analyzed in [13,14,9]. Using geometric relationships to bound the evolution, we determine necessary and sufficient conditions on the parameters involved which will lead to stability from any/some initial conditions.

### *1.2 Paper Organization*

The paper is organized as follows. The problem formulation and the proposed approach is described in Section 2. Conditions for the existence of equilibria together with an insightful geometric interpretation are presented in Section 3. Convergence results for the IBR algorithm are derived in Section 4 along with supporting numerical results. Finally, conclusions and directions for future research are discussed in Section 5. Proofs of the results are available at **??**.

## 2 Problem Formulation

The problem scenario is depicted in Figure 1. This work assumes that the data and the computation to be carried out are such that it is possible for the sensor to compute an estimate of the computation locally using some random subset of the data, and that the statistics (up to the mean with a finite second moment) about how the actual value is distributed given a value of estimate is known to the sensor.

For example, suppose that the data $d \in \mathbb{R}^{N \times M}$, consisting of $N$ data points each represented by an $M$-dimensional feature vector, is uploaded through a trusted sensor to the third-party computer. During the upload process, the sensor could retain a randomly sparsified sample $\hat{d} \in \mathbb{R}^{N \times M}$ of the original data $d$. The data, once stored on the central computer, can be compromised by the attacker leading to a different value $\bar{d} \in \mathbb{R}^{N \times M}$. The sensor is interested to compute the true value of the function $y = g(d) \in \mathbb{R}^k$ of the data $d$ where $g : \mathbb{R}^{N \times M} \to \mathbb{R}^k$ is an algorithm of interest. For example, $g(d) \in \mathbb{R}$ could be the maximum singular value of a matrix with $\hat{d}$ being a sparsified sample of $d$. The sensor has only an approximate knowledge, $\hat{y} = g(\hat{d}) \in \mathbb{R}^k$, of $y$, obtained from the sparsified data $\hat{d}$. The third-party computer computes the value $\bar{y} = g(\bar{d}) \in \mathbb{R}^k$ based on the corrupted data $\bar{d}$. This paper does not address the problem of how to construct a sparsified sample $\hat{d}$, but rather makes an implicit assumption that for a given number of non-zero elements in $\hat{d}$ and a corresponding sparsification procedure, the distribution of $y$ given $\hat{y}$ can be characterized a priori. The actual construction of a specific sampling procedure for a computation such as the maximum eigenvalue would be a topic of future investigation and we direct an interested reader to [1] and [5] for related existing results.

The sensor needs to design a fusion function $\phi : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^k$ that provides $y_{\text{fused}}$ so that the effect of the attack is minimized, i.e., the sensor seeks to minimize the squared error between $y_{\text{fused}}$ and $y$ in an expected sense. On the other hand, the attacker seeks to drive the fused value to be close to a different value $y_A$ instead of the true $y$, and therefore, seeks to minimize the squared error between $y_{\text{fused}}$ and $y_A$ in an expected sense. The attacker returns a value $\bar{y} \in \mathbb{R}^k$, which will be chosen in order to minimize the mean squared error between $y_{\text{fused}}$ and $y_A$.

Given a value of $\hat{y}$, one can view the true value $y$ as a random variable, and we assume that the conditional distribution of $y$ given $\hat{y}$ $(y|\hat{y})$ is known to the sensor. From the sensor's perspective, this work only requires it to know the mean of $y$ given $\hat{y}$ and the fact that $y$ given $\hat{y}$ has bounded variance. This paper proposes a new protocol between sensor and the computer that will enable a sensor to compute an optimal fusion strategy $y_{\text{fused}} = \phi(\hat{y}, \bar{y}) \in \mathbb{R}^k$, in the space of convex combinations of the approximate value $\hat{y}$ and the third-party computer's output $\bar{y}$, to minimize the mean square error, namely,

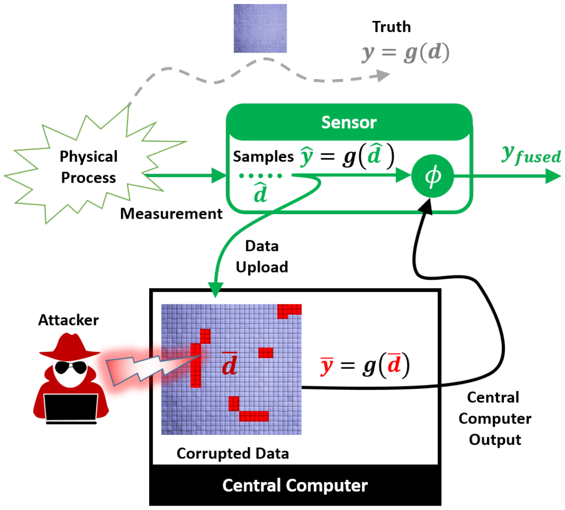

Fig. 1. Trusted Computation over an adversarial third-party computer. The sensor has a large dataset out of which it draws a sample $\hat{d}$ and uploads the entire dataset onto the central computer. The computer returns some value $\bar{y}$ as the output of the computation, while the sensor computes a value $\hat{y}$ based on the sample $\hat{d}$. The goal is to fuse, using a function $\phi(.)$, the two quantities $\hat{y}$ and $\bar{y}$ in an optimal manner.

$$J_D := \mathbb{E}_y[\|y_{\text{fused}} - y\|^2 \,|\, \hat{y}] \,, \tag{1}$$

where the norm $\|\cdot\|$ is the 2-norm. The goal of the attacker is to pick a $\bar{y}$ to minimize a different mean square error, i.e.,

$$J_A := \mathbb{E}_{\hat{y}}[\|y_{\text{fused}} - y_A\|^2] \,, \tag{2}$$

where $y_A$ is a value that the attacker chooses to corrupt the sensor. We will assume that $y_A$ is a fixed and non-random value known only to the attacker. In other words, the goal of the attacker is to choose a value $\bar{y}$ to be given to the sensor as the *output of the computation* so that when the sensor fuses this value with its private estimate $\hat{y}$, the fused value $y_{\text{fused}}$ becomes close to a certain value $y_A$ which is selected by the attacker. The value $y_A$ is the value that the attacker wants the sensor to believe is the true output. The attacker may decide to tamper with the raw data to compute its $\bar{y}$, in which case it modifies $d$ to $\bar{d}$ such that $\bar{y} = g(\bar{d})$. Alternatively, it may directly tamper the correct output coming from the central computer, in which case it simply replaces the correct value with its own value $\bar{y}$.

The mismatch in the information structure available to both players is reflected in their respective cost functions in that the random variable relative to which the expectation is computed is the one whose realization is not known to that player.

In the space of allowed strategies, we choose the function $\phi(.) = \phi_\alpha(.)$ to be a convex combination of $\hat{y}$ and $\bar{y}$, namely

$$y_{\text{fused}} := \phi_\alpha(\hat{y}, \bar{y}) = \alpha\hat{y} + (1-\alpha)\bar{y}\,. \tag{3}$$

In other words, the sensor needs to compute an $\alpha \in [0,1]$, thus deciding the weight to attach to its own value $\hat{y}$ and the attacker's output $\bar{y}$. Therefore, both cost functions, $J_D$ and $J_A$ are functions of the action $\alpha \in [0,1]$ of the sensor and the action $\bar{y} \in \mathbb{R}^k$ of the attacker. To be specific, let $\mu$ denote the expected value of $y$ given $\hat{y}$ using a density function chosen by the sensor, modeling the sensor's knowledge about the data, and let $\zeta$ denote the expected value of $\hat{y}$ using the density function chosen by the attacker, modeling the knowledge the attacker has about the data. In the space of linear strategies (3), the cost functions (1) simplifies to:

$$\begin{aligned} J_D(\alpha, \bar{y}) &= \int_{\mathbb{R}^k} \|\alpha\hat{y} + (1-\alpha)\bar{y} - w\|^2 f_{y|\hat{y}}(w)dw \\ &= \|\alpha(\hat{y}-\bar{y}) + \bar{y}\|^2 + \int_{\mathbb{R}^k} \|w\|^2 f_{y|\hat{y}}(w)dw \\ &\quad - 2(\alpha(\hat{y}-\bar{y}) + \bar{y})^T \underbrace{\int_{\mathbb{R}^k} w f_{y|\hat{y}}(w)dw}_{\mu}\,, \end{aligned} \tag{4}$$

where $f_{x|y}(\cdot)$ denotes the probability density function of a random variable $x$ given $y$. The cost function (2) simplifies to:

$$\begin{aligned} J_A(\alpha, \bar{y}) &= \int_{\mathbb{R}^k} \|\alpha w + (1-\alpha)\bar{y} - y_A\|^2 f_{\hat{y}}(w)dw \\ &= \alpha^2 \int_{\mathbb{R}^k} \|w\|^2 f_{\hat{y}}(w)dw + \|(1-\alpha)\bar{y} - y_A\|^2 \\ &\quad + 2\alpha((1-\alpha)\bar{y} - y_A)^T \underbrace{\int_{\mathbb{R}^k} w f_{\hat{y}}(w)dw}_{\zeta}\,. \end{aligned} \tag{5}$$

This is a non zero-sum game for which we will consider the following notion of equilibrium.

**Definition 1 (Equilibrium)** *An admissible pair* $(\alpha^*, \bar{y}^*)$ *is an* equilibrium *if*

$$\begin{aligned} J_D(\alpha^*, \bar{y}^*) &\le J_D(\alpha, \bar{y}^*)\,, \qquad \forall \alpha \in [0,1],\ \textit{and} \\ J_A(\alpha^*, \bar{y}^*) &\le J_A(\alpha^*, \bar{y})\,, \qquad \forall \bar{y} \in \mathbb{R}^k\,. \end{aligned}$$

*Further, if* $\alpha^* \in (0,1)$*, then the resulting equilibrium pair is said to be* mixed. □

In other words, a pair of strategies is in equilibrium if no other strategy can give a *strictly better* cost against the opponent's strategy, from each player's perspective. Further, when the best response of the sensor is *mixed*, it means that the sensor selects a non-trivial weighted combination of $\hat{y}$ and $\bar{y}$. However, it will be clear from the next section that an explicit one-shot computation of the equilibrium strategies requires the players to have full knowledge of all the problem parameters, i.e., $\hat{y}$, $y$, $y_A$, and the expected values $\mu$ and $\zeta$ of $y$ given $\hat{y}$ and $y$, respectively. Clearly, from one player's point of view, this information is not available. Indeed, $y_A$ is a private information that the attacker has and is not shared with the sensor and the $\hat{y}$ is private information of the sensor which is not shared with the attacker. Therefore, we will consider the following iterative scheme in which each player will announce its best response to a strategy announced by the opponent, with the attacker playing first. This protocol is summarized in Algorithm 1.

**Algorithm 1** Iterated Best Response

```
Assumes: Attacker plays first, i.e., select a value for $\bar{y}_0$.
$i = 0$
$\alpha_i = 0$
# Exit when $\alpha$ reaches steady-state or becomes 1
while $\alpha_i$ sequence has not converged do
  $i = i + 1$
  # Client updates its action based on $\bar{y}_{i-1}$
  $\alpha_i = \text{argmin}_{\alpha\in[0,1]} J_D(\alpha, \bar{y}_{i-1})$.
  # Attacker updates its action based on $\alpha_i$
  $\bar{y}_i = \text{argmin}_{\bar{y}} J_A(\alpha_i, \bar{y})$.
  # Exit and trust the sensor output only
  if $\alpha_i == 1$ then
    return $\alpha_i$
  end if
end while
return $\alpha_i$ and $\bar{y}_i$
```

To apply Algorithm 1, we will first need to compute the best responses of each player against an action of the opponent. Setting $\partial J_D / \partial\alpha = 0$ in (4), we obtain the unconstrained minimizer

$$\begin{aligned} &(\alpha(\hat{y}-\bar{y}) + \bar{y})^T(\hat{y}-\bar{y}) - (\hat{y}-\bar{y})^T\mu = 0 \\ \Leftrightarrow\ &\alpha^*_{\text{unc}}(\bar{y}) = \frac{(\mu-\bar{y})^T(\hat{y}-\bar{y})}{\|\bar{y}-\hat{y}\|^2}\,. \end{aligned}$$

Due to the constraint $\alpha^* \in [0,1]$, the best response for the sensor is:

$$\alpha^*(\bar{y}) = \begin{cases} 0\,, & \text{if } (\bar{y}-\mu)^T(\hat{y}-\bar{y}) \ge 0\,, \\ 1\,, & \text{if } (\hat{y}-\mu)^T(\bar{y}-\hat{y}) \ge 0\,, \\ \dfrac{(\bar{y}-\mu)^T(\bar{y}-\hat{y})}{\|\bar{y}-\hat{y}\|^2}, & \text{otherwise.} \end{cases} \tag{6}$$

A similar calculation yields the best response for the attacker:

$$\bar{y}^*(\alpha) = \begin{cases} \dfrac{y_A - \alpha\zeta}{1-\alpha}, & \text{if } \alpha \ne 1\,, \\ \text{any value}\,, & \text{if } \alpha = 1\,. \end{cases} \tag{7}$$

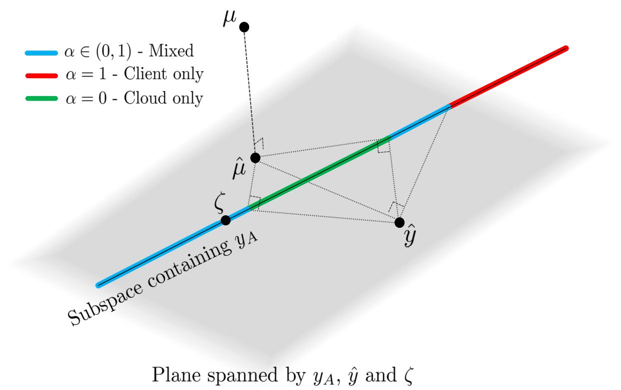
$\mu$
$\alpha \in (0,1)$ - Mixed
$\alpha = 1$ - Client only
$\alpha = 0$ - Cloud only
$\hat{\mu}$
$\zeta$
$\hat{y}$
Subspace containing $y_A$
Plane spanned by $y_A$, $\hat{y}$ and $\zeta$

Fig. 2. The locations for $\bar{y}$ (in the subspace containing $y_A$) that lead to different values of $\alpha^*$, given the values of $y_A$, $\hat{y}$ and $\mu$. The green locations will lead to $\alpha = 0$, the red locations will lead to $\alpha = 1$ and the blue locations will lead to a mixed $\alpha$.

Observe that $\bar{y}^*$ is a linear combination of $y_A$ and $\zeta$. Since we have a total of four parameters $\hat{y}, \zeta, y_A$ and $\mu$, it is convenient considering the plane containing the three points $\hat{y}$, $\zeta$, $y_A$ and let $\hat{\mu}$ be the *orthogonal projection* of $\mu$ on to this plane. Therefore, we can write $\mu := \hat{\mu} + \mu^{\perp}$. Since ${\mu^{\perp}}^T(\bar{y} - \hat{y}) = 0$, the expression for $\alpha^*(\bar{y})$ can be rewritten as

$$\alpha^*(\bar{y}) = \begin{cases} 0\,, & \text{if } (\bar{y} - \hat{\mu})^T(\hat{y} - \bar{y}) \geq 0\,, \\ 1\,, & \text{if } (\hat{y} - \hat{\mu})^T(\bar{y} - \hat{y}) \geq 0\,, \\ \dfrac{(\bar{y} - \hat{\mu})^T(\bar{y} - \hat{y})}{\|\bar{y} - \hat{y}\|^2}\,, & \text{otherwise.} \end{cases} \tag{8}$$

Interestingly, we can give a geometric interpretation to (8) and (7), as shown in Figure 2. In particular, the figure illustrates the locations of $\bar{y}$ that lead to different values of $\alpha^*$, given the values of $y_A$, $\hat{y}$ and $\hat{\mu}$. In the plane defined by $y_A$, $\hat{y}$ and $\zeta$, a given subspace containing $y_A$ can be divided into at most three distinct regions corresponding to the three regimes in (8). The green regime corresponds to the set of points $\bar{y}$ for which $\alpha^*(\bar{y}) = 0$ and when it exists, it lies between two blue regimes corresponding to the set of points $\bar{y}$ for which $\alpha^*(\bar{y}) \in (0, 1)$. The red regime corresponds to the set of points $\bar{y}$ for which $\alpha^*(\bar{y}) = 1$. Note that the boundary points between the green and blue regimes satisfy the property that the lines joining them to $\hat{\mu}$ and $\hat{y}$ are orthogonal to each other, as highlighted in Figure 2.
The termination condition within the if loop of Algorithm 1 is due to the fact that when $\alpha = 1$, it implies that the output of the central computer is not to be trusted, and thereafter, the computer is not providing any further useful information than the sensor's private value of $\hat{y}$. In order to discard trivial initial conditions for which Algorithm 1 would exit at the first iteration, we will restrict our discussion to *non-trivial* initial values of $\bar{y}_0$ defined next.

**Definition 2 (Non-trivial initial condition)** *An initial condition $\bar{y}_0$ is said to be* non-trivial *if Algorithm 1 does not exit at the first iteration.*

We will consider the following two notions of convergence for Algorithm 1.

**Definition 3 (Weak Convergence)** *Algorithm 1 is said to possess* weak convergence *property if, for* some *non-trivial initial condition $\bar{y}_0$ and for some values of the means $\mu$ and $\zeta$, Algorithm 1 outputs the final value of $\alpha^* \in [0, 1)$.*

**Definition 4 (Strong Convergence)** *Algorithm 1 is said to possess* strong convergence *property if, for* every *non-trivial choice of $\bar{y}_0$, and for every value of the means $\mu$ and $\zeta$, Algorithm 1 outputs the final value of $\alpha^* \in [0, 1)$.*

In the sequel, we will see that the weak convergence concept will be useful from the sensor's perspective, whereas strong convergence will be useful from the attacker's perspective.
The main contributions in the rest of this paper are to present conditions on the problem parameters, viz. $\hat{y}$ ,$y$, $y_A$, $\mu$ and $\zeta$, under which: 1) equilibrium strategies exist for Algorithm 1 and 2) Algorithm 1 demonstrates weak or strong convergence properties. Whenever Algorithm 1 converges, then the *steady-state* strategies actually correspond to an equilibrium in the sense of Definition 1.

## 3 Equilibrium Strategies

In this section, we will derive conditions on the parameters $\hat{y}, y_A, \mu$ and $\zeta$ under which equilibria exist for the system of equations (6) and (7).
Let $\delta := \zeta - \mu$, $z_A := y_A - \zeta$, and $\hat{z} := \hat{y} - \zeta$. The following is the main result on the equilibria of the above system.

**Theorem 3.1 (Equilibrium)** *For the system* (6) *and* (7)*, we have the following:*

*i. The pair of strategies $(\alpha^*, \bar{y}^*) = (0, y_A)$ is an equilibrium if and only if $(y_A - \mu)^T(\hat{y} - \mu) \geq \|y_A - \mu\|^2$.*
*ii. An equilibrium in mixed strategies exists if and only if $(\hat{z}^T(\hat{z} + 2z_A - \delta))^2 \geq 4z_A^T(\hat{z} + z_A - \delta)\hat{z}^T\hat{z}$.*

This result provides conditions on the game parameters under which one of the two types of equilibria considered in this paper would exist. The first type is the one when $\alpha^* = 0$, i.e., the sensor completely trusts the output from the central computer. The condition in (i) essentially describes the set of values for the attacker's intent $y_A$ for which it will be beneficial for the sensor to use the output $\bar{y}$ from the central computer than its own value $\hat{y}$. The second type is the one which will require the sensor to *fuse* its private value $\hat{y}$ with the value $\bar{y}$ in a non-trivial way ($\alpha^* \in (0, 1)$).
**Proof:** The best response of the players for the case when $\alpha^* = 0$ yields $\bar{y}^* = y_A$. Conversely, if $\bar{y}^* = y_A$,

then $\alpha^* = 0$ if and only if

$$\begin{aligned} &(y_A - \mu)^T(\hat{y} - y_A) \geq 0 \\ &\Leftrightarrow (y_A - \mu)^T(\hat{y} - \mu - (y_A - \mu)) \geq 0\,, \end{aligned}$$

which establishes the first claim.
For the second claim, since we are searching for mixed policies $\alpha^*$, we substitute the expression for $\alpha^*(\bar{y})$ into the fixed point equation for $\bar{y}$ to obtain

$$\bar{y} = \frac{y_A\|\hat{y} - \bar{y}\|^2}{(\hat{y} - \mu)^T(\hat{y} - \bar{y})} + \frac{(\mu - \bar{y})^T(\hat{y} - \bar{y})}{(\mu - \hat{y})^T(\hat{y} - \bar{y})}\zeta \qquad (9)$$

Subtracting $\zeta$ from both sides, we have

$$\begin{aligned} \bar{y} - \zeta &= \frac{y_A\|\hat{y} - \bar{y}\|^2}{(\hat{y} - \mu)^T(\hat{y} - \bar{y})} + \Big(\frac{(\mu - \bar{y})^T(\hat{y} - \bar{y})}{(\mu - \hat{y})^T(\hat{y} - \bar{y})} - 1\Big)\zeta \\ \bar{y} - \zeta &= \frac{y_A\|\hat{y} - \bar{y}\|^2}{(\hat{y} - \mu)^T(\hat{y} - \bar{y})} - \frac{\|\hat{y} - \bar{y}\|^2}{(\mu - \hat{y})^T(\hat{y} - \bar{y})}\zeta. \end{aligned}$$

By denoting $\bar{z} := \bar{y} - \zeta$, we obtain

$$\bar{z} = \frac{\|\hat{z} - \bar{z}\|^2}{(\hat{z} - \delta)^T(\hat{z} - \bar{z})} z_A \;\Rightarrow\; \bar{z}^* = r z_A\,,$$

where $r$ is a scalar that must satisfy

$$r = \frac{\|\hat{z} - r z_A\|^2}{(\hat{z} - r z_A)^T(\hat{z} - \delta)}\,.$$

On simplifying, we obtain the following quadratic equation in $r$:

$$z_A^T(\hat{z} + z_A - \delta)r^2 - \hat{z}^T(\hat{z} + 2z_A - \delta)r + \hat{z}^T\hat{z} = 0\,.$$

The condition now follows from the existence of real roots to the above quadratic equation. □

Clearly, the computation of equilibria requires complete information of the problem parameters. Therefore, in the next sub-section, we will characterize conditions on the parameters under which Algorithm 1 will converge.

## 4 Convergence Analysis

In this section, we will derive conditions under which Algorithm 1 possesses weak and strong convergence.
From the point of view of the sensor, the goal is to characterize conditions on the attack parameter $y_A$ for which Algorithm 1 will converge in the weak and in the strong sense. However, there is an additional uncertainty on where the attacker's mean $\zeta$ will lie. We will commence the convergence analysis for a given value of $\zeta$ and then extend it to the case when there is a bound on the amount of mismatch, such as one in the following assumption.

**Assumption 4.1 (Mismatch parameter)** *The mismatch $\zeta - \mu$ satisfies*

$$\|\zeta - \mu\| \leq \frac{\epsilon}{1+\epsilon}\|\hat{y} - \mu\|\,,$$

*for some given value of $\epsilon \in [0, 1]$.*

This assumption is reasonable to expect because the deviation $\|\hat{y} - \mu\|$ will be relatively large when only a small subset of the data being sampled. The convergence analysis will require two intermediate results using geometry which we present in the next sub-section.

### *4.1 Preliminary Geometric Results*

Given any $y_A$, we will first show that Assumption 4.1 leads to the following upper bound on $\|\zeta - \hat{\mu}\|$, where $\hat{\mu}$ is the orthogonal projection of $\mu$ on to the plane containing $\zeta, y_A$ and $\hat{y}$.

**Lemma 4.2 (Mismatch bound)** *For given values of $\zeta, y_A$ and $\hat{y}$, under Assumption 4.1, we have $\|\zeta - \hat{\mu}\| \leq \epsilon\|\hat{y} - \hat{\mu}\|$.*

We will also require another geometric result which will aid the proof of the necessary condition. We introduce the following notation: given two vectors $x_1, x_2 \in \mathbb{R}^k$, the vector $\mathcal{P}(x_1, x_2) \in \mathbb{R}^k$ denotes the orthogonal projection of $x_1$ onto $x_2$.

**Lemma 4.3** *Suppose that a point $\bar{y}$ satisfies:*
*i. $\alpha^*(\bar{y}) \in (0, 1)$, and*
*ii. $\bar{y}$ lies in the closure of the half plane defined by the line joining $\hat{y}$ and $y_A$ with the side not containing $\hat{\mu}$.*
*Then, $\|\,\mathcal{P}(\hat{y} - \hat{\mu}, y_A - \hat{y})\| \geq \|\,\mathcal{P}(\hat{y} - \hat{\mu}, \bar{y} - \hat{y})\|$.*

### *4.2 Weak Convergence*

We now begin our analysis of a necessary condition assuming that $y_A$ and $\zeta$ are known, which therefore implies that $\hat{\mu}$ is known.

**Theorem 4.4 (Weak Convergence given $\zeta, y_A$)** *For given values of $\zeta, \hat{y}$ and $y_A$, the following hold:*
*i. For Algorithm 1 to converge to $(\alpha^* = 0, \bar{y}^* = y_A)$, $y_A$ must satisfy $(y_A - \hat{\mu})^T(\hat{y} - y_A) \geq 0$.*
*ii. Under Assumption 4.1, for Algorithm 1 to weakly converge, $y_A$ must satisfy $(\hat{y} - \hat{\mu})^T(y_A - \hat{y}) < 0$.*
*iii. Let $\Psi_{\hat{\mu}}$ be defined as the closure of the half-plane defined by the line joining $\hat{y}$ and $y_A$ and which contains the point $\hat{\mu}$. For Algorithm 1 to yield a converging sequence of mixed $\alpha^* \in (0, 1)$, $y_A$ must satisfy either one of the following:*
*a. $\|y_A - \zeta\| \leq \|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - y_A)\|\,, \quad \zeta \in \Psi_{\hat{\mu}}$,*
*b. $\|y_A - \zeta\| \leq \|\hat{y} - \hat{\mu}\|\,, \quad \zeta \notin \Psi_{\hat{\mu}}$.*

**Proof:** Algorithm 1 results into a steady-state value $(\alpha^* = 0, \bar{y}^* = y_A)$ only if (cf. (6) and (7)),

$$(y_A - \mu)^T(\hat{y} - y_A) \geq 0 \Leftrightarrow (y_A - \hat{\mu})^T(\hat{y} - y_A) \geq 0\,,$$

since $\hat{\mu}$ is the orthogonal projection of $\mu$ on to the plane containing $\zeta, y_A$ and $\hat{y}$. This proves the first case.

To prove the second case, suppose that Algorithm 1 converges to an $\alpha^* \in [0, 1)$ from some arbitrary, non-trivial initial condition $\bar{y}_0$. If $\bar{y}_0$ is in the green region ($\alpha_0 = 0$), then after one iteration of Algorithm 1, we obtain $\bar{y}_1 = y_A$. If $y_A$ is also in the green region (i.e., corresponding to $\alpha_1 = 0$), then the algorithm converges to an equilibrium with $\alpha^* = 0$ and case i) applies. On the other hand, if $y_A$ is in the blue region ($\alpha \in (0, 1)$), then it follows that $\bar{y}_1 = y_A$, and thus $\bar{y}_1$ lies in the blue region ($\alpha \in (0, 1)$). Therefore, we can assume that without any loss of generality, $\bar{y}_0$ is in the blue region. From (8), it needs to hold that $(\hat{y} - \hat{\mu})^T(\hat{y} - \bar{y}_0) > 0$. Now, substituting (7) into (8) and on following the same steps that led to (9), we have that after one iteration of Algorithm 1, $\bar{y}_1$ is such that the vector $\bar{y}_0 - \zeta$ is a *positive scalar multiple* of the vector $y_A - \zeta$. More precisely, we have

$$\bar{y}_1 - \zeta = \frac{\|\hat{y} - \bar{y}_0\|^2}{(\hat{y} - \hat{\mu})^T(\hat{y} - \bar{y}_0)}(y_A - \zeta)\,.$$

Now, observe that the ratio

$$\frac{\|\hat{y} - \bar{y}_0\|^2}{(\hat{y} - \hat{\mu})^T(\hat{y} - \bar{y}_0)} = \frac{\|\hat{y} - \bar{y}_0\|}{\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_0)\|} > 1\,, \quad (10)$$

which implies that for any non-trivial $\bar{y}_0$, $\bar{y}_1 - \zeta$ is a positive scalar multiple *greater than unity* of $y_A - \zeta$.

Let us assume, for the sake of arriving at a contradiction, that $0 \leq (\hat{y} - \hat{\mu})^T(y_A - \hat{y})$, namely that the condition of case ii) does not hold. Therefore, we can write

$$\begin{aligned}
&0 \leq (\hat{y} - \hat{\mu})^T(y_A - \hat{y}) \\
\Rightarrow &0 \leq (\hat{y} - \hat{\mu})^T(y_A - \zeta + \zeta - \hat{\mu} + \hat{\mu} - \hat{y}) \\
\Rightarrow &\|\hat{y} - \hat{\mu}\|^2 - \|\hat{y} - \hat{\mu}\|\|\zeta - \hat{\mu}\|\cos\psi \leq (\hat{y} - \hat{\mu})^T(y_A - \zeta)\,,
\end{aligned}$$

where $\psi$ is the smaller angle between the vectors $\hat{y} - \hat{\mu}$ and $\zeta - \hat{\mu}$. Since Assumption 4.1 holds, Lemma 4.2 holds. Therefore, the left hand side of the above inequality is non-negative, and thus,

$$0 \leq (\hat{y} - \hat{\mu})^T(y_A - \zeta) \Rightarrow 0 \leq (\hat{y} - \hat{\mu})^T(\bar{y}_1 - \zeta), \quad (11)$$

since $\bar{y}_1 - \zeta$ is a positive scalar multiple of $y_A - \zeta$.

Now, from (10), we can write

$$\begin{aligned}
0 &\leq (\hat{y} - \hat{\mu})^T(y_A - \hat{y}) \\
&= (\hat{y} - \hat{\mu})^T\left(\frac{1}{r}(\bar{y}_1 - \zeta) + + (\zeta - \hat{y})\right) \\
&\leq (\hat{y} - \hat{\mu})^T(\bar{y}_1 - \hat{y})\,,
\end{aligned}$$

where the second inequality follows from (11) and from the fact that $r = \|\hat{y} - \bar{y}_0\|^2/((\hat{y} - \hat{\mu})^T(\hat{y} - \bar{y}_0)) \geq 1$. This however implies, from (8), that $\bar{y}_1$ is in the red region ($\alpha_1 = 1$). But this is not possible as we already showed that $\bar{y}_1$ is in the blue region ($\alpha = (0, 1)$). This proves the second case.

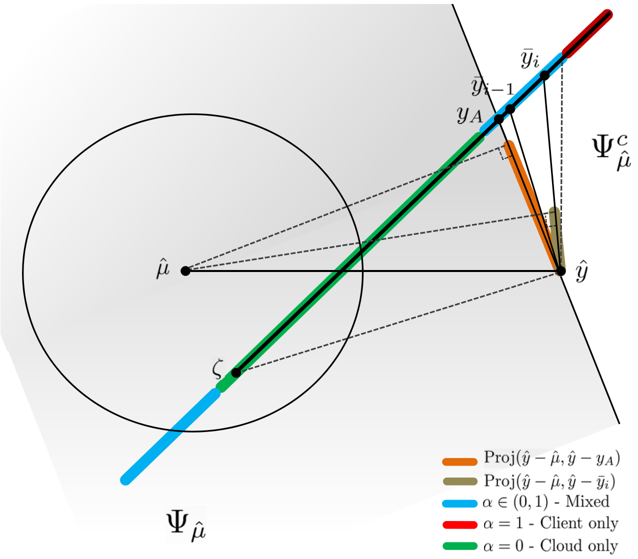
$\bar{y}_i$
$\bar{y}_{i-1}$
$y_A$
$\Psi^c_{\hat{\mu}}$
$\hat{\mu}$
$\hat{y}$
$\zeta$
$\Psi_{\hat{\mu}}$
Proj$(\hat{y} - \hat{\mu}, \hat{y} - y_A)$
Proj$(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_i)$
$\alpha \in (0, 1)$ - Mixed
$\alpha = 1$ - Client only
$\alpha = 0$ - Cloud only

Fig. 3. Illustrating one possible scenario in the proof of Theorem 4.4. In this scenario, the points $\zeta$ and $\hat{\mu}$ lie on the same side of the line joining $y_A$ and $\hat{y}$, i.e., the requirement in the third case in the statement of Theorem 4.4 is satisfied, which leads to inequality (14) to hold. In gray, we indicate $\Psi_{\hat{\mu}}$, the closure of the half-plane defined by the line joining $\hat{y}$ and $y_A$ and which contains the point $\hat{\mu}$.

To prove the last case, without loss of generality, we can assume that $(\bar{y}_0 - \zeta)$ is a positive scalar multiple of $y_A - \zeta$, and that $\bar{y}_0$ is in the blue region (i.e., corresponding to $\alpha \in (0, 1)$), using the previous case.

Suppose that for every $i \geq 0$, $\bar{y}_i$ is in the blue region, i.e., every $\alpha_i \in (0, 1)$. Then consider the recursion,

$$\begin{aligned}
\|\bar{y}_i - \zeta\| &= \frac{\|\hat{y} - \bar{y}_{i-1}\|^2\|y_A - \zeta\|}{|(\hat{y} - \hat{\mu})^T(\hat{y} - \bar{y}_{i-1})|} \\
\Leftrightarrow \|\bar{y}_i - \zeta\| &= \frac{\|\hat{y} - \bar{y}_{i-1}\|\|y_A - \zeta\|}{\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_{i-1})\|}\,, \qquad (12)
\end{aligned}$$

Note that (12) may be viewed as a *discrete-time dynamical system* in $\bar{y}$. Replacing $\bar{y}_0$ by $\bar{y}_{i-1}$ in (10), we have that for every $i \geq 1$, and for any $\bar{y}_0$, such that $\bar{y}_0 - \zeta$ is a positive scalar multiple of $y_A - \zeta$,

$$\|\bar{y}_{i-1} - \zeta\| > \|y_A - \zeta\|\,. \qquad (13)$$

We need now to distinguish two cases.

a. If $\zeta \in \Psi_{\hat{\mu}}$, then (13) implies that for every $i \geq 1$, the point $\bar{y}_{i-1} \in \Psi^c_{\hat{\mu}}$, namely the complement of $\Psi_{\hat{\mu}}$. We refer to Figure 3 for a geometric interpretation of the proof. Additionally, we have that $\alpha^*(\bar{y}_{i-1}) \in (0, 1)$. Thus, applying Lemma 4.3, we conclude that for every $i \geq 1$,

$$\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_{i-1})\| \leq \|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - y_A)\|\,. \quad (14)$$

Further, from applying the triangle inequality to the three points $\bar{y}_{i-1}, \hat{y}$ and $\zeta$, we have

$$\|\hat{y} - \bar{y}_{i-1}\| \geq \|\bar{y}_{i-1} - \zeta\| - \|\hat{y} - \zeta\| \,. \tag{15}$$

Combining this together with (14) and (12) yields

$$\|\bar{y}_i - \zeta\| \geq \frac{\|y_A - \zeta\|(\|\bar{y}_{i-1} - \zeta\| - \|\hat{y} - \zeta\|)}{\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - y_A)\|} \,,$$

which is a linear system in the quantity $\|\bar{y}_i - \zeta\|$. This implies that Algorithm 1 will output a converging sequence of mixed $\alpha^*$'s only if $\|y_A - \zeta\| \leq \|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - y_A)\|$.

b. If $\zeta \in \Psi^c_{\hat{\mu}}$, then we can apply the following upper bound $\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_i)\| \leq \|\hat{y} - \hat{\mu}\|$, which follows from the fact that the length of the projection of a vector $x$ onto any another vector can never exceed the length of $x$ itself. Following the same steps as in the previous case, we conclude that the sequence $\|\bar{y}_i - \zeta\|$ will converge only if $\|y_A - \zeta\| \leq \|\hat{y} - \hat{\mu}\|$. This proves the iii) case.

□

We numerically verify this result by studying the region of divergence for Algorithm 1 in two dimensions. In the planar case, the value of the mean $\mu = \hat{\mu} = (0, 0)$, and the sensor's value $\hat{y} = (0.8, 0)$. The attacker's mean, $\zeta = (0.3, -0.2)$. For every value of $y_A$ on a grid in the neighborhood of $\hat{\mu}$, we ran Algorithm 1 for a set of different initial values $\bar{y}_0$. The results are summarized in Figure 4. If the algorithm converged [1] to an $\alpha^* \in [0, 1)$ for some choice of $\bar{y}_0$, then the corresponding point $y_A$ is shown as a (green) dot. Otherwise, it is shown as a (red) cross. The analytically derived necessary conditions from Theorem 4.4 is shown as a black dashed contour.

We now revisit the fact that the parameter $\zeta$ is not known to the sensor. Suppose that $\zeta \in \mathcal{C}$, where $\mathcal{C}$ is a set known to the sensor and which satisfies Assumption 4.1. Define the set $N_\zeta$ as the set of all points $x \in \mathbb{R}^k$ which satisfy the necessary conditions from Theorem 4.4. Then, the following result holds.

**Corollary 4.5 (Necessary Condition)** *Suppose that $\mathcal{C}$ is a set of all $\zeta \in \mathbb{R}^k$ that satisfies Assumption 4.1. Then, a necessary condition for weak convergence of Algorithm 1 is that $y_A \in \bigcup_{\zeta \in \mathcal{C}} N_\zeta$.*

In Figure 5 we plot the analytic condition from Corollary 4.5 for different sets $\mathcal{C}$ in $\mathbb{R}^2$. Since this example is planar, $\hat{\mu} = \mu$. In particular, let the value of the mean $\mu = (0, 0)$, the sensor's value $\hat{y} = (1, 0)$, and the set $\mathcal{C}$ be different circles of increasing radii around $\mu$. Figure 5 shows how the set $\bigcup_{\zeta \in \mathcal{C}} N_\zeta$ evolves with increasing radius, $\delta$, of $\mathcal{C}$. As is expected, the set computed for a higher values of the radius contains the set computed for a lower one. In other words, for a higher level of uncertainty about the attacker's mean $\zeta$, the necessary condition becomes more conservative.

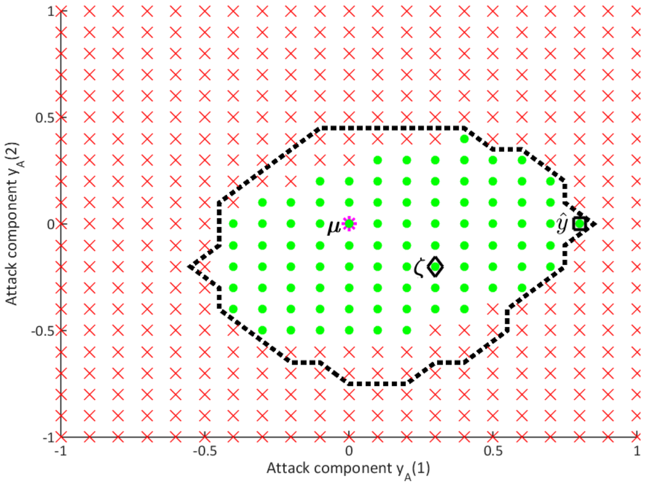

Fig. 4. Numerically generated plot to study the region of divergence for Algorithm 1 in two dimensions. The value of the mean $\hat{\mu} = (0, 0)$ (shown as a (magenta) star), the attacker's mean $\zeta = (0.3, -0.2)$ (shown as a (black) diamond) and the sensor's value $\hat{y} = (0.8, 0)$, shown as a ( black) square. For every value of $y_A = [y_A(1), y_A(2)]$ on a grid in the neighborhood of $\mu$, we run Algorithm 1 for a set of different initial values $\bar{y}_0$. If the algorithm converges to an $\alpha^* \in [0, 1)$ for some choice of $\bar{y}_0$, then the corresponding point $y_A$ is shown as a (green) dot. Otherwise, it is shown as a (red) cross. The analytically derived necessary conditions from Theorem 4.4 is shown as a black dashed contour.

### 4.3 Strong Convergence

The next result establishes a sufficient condition for strong convergence.

**Theorem 4.6 (Strong Convergence given $\zeta, y_A$)** *For given values of $\zeta, \hat{y}$ and $y_A$, Algorithm 1 possesses strong convergence if*

*i.* $$(y_A - \zeta)^T(\hat{y} - \hat{\mu}) \leq 0 \,, \text{ and} \tag{16}$$

*ii.* $$\|y_A - \zeta\| \leq \min(\|\,\mathcal{P}(\hat{y} - \hat{\mu}, y_A - \zeta)\|, \|\,\mathcal{P}(\hat{y} - \hat{\mu}, y_A - \hat{y})\|) \,. \tag{17}$$

*Additionally, condition* (16) *is also necessary for strong convergence.*

**Proof:** Following the same steps that lead to (13) in the proof of Theorem 4.4, we can assume without any loss of generality that $\bar{y}_0 - \zeta$ is a positive scalar multiple (greater than unity) of $y_A - \zeta$. Recall (12):

$$\|\bar{y}_i - \zeta\| = \frac{\|\hat{y} - \bar{y}_{i-1}\|}{\|\,\mathcal{P}(\hat{y} - \hat{\mu}, \hat{y} - \bar{y}_{i-1})\|} \|y_A - \zeta\| \,. \tag{18}$$

[1] The convergence condition in Algorithm 1 was approximated by running the while loop for a sufficiently large number of iterations. In our simulations, we used 100 steps.

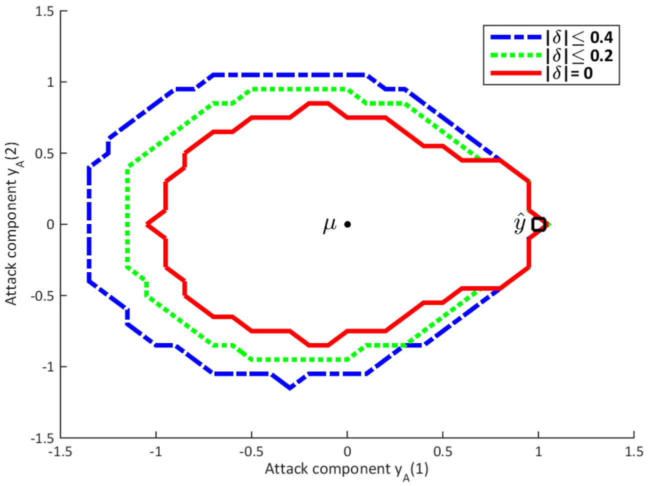

Fig. 5. Plot of how the analytic necessary condition from Corollary 4.5 evolves for increasing radii of the set $\mathcal{C}$ which contains $\zeta$. This plot has been numerically generated by sampling 100 points uniformly randomly out of circles $\mathcal{C}$ of radii $\delta$ equal to zero (solid (red) line), 0.2 (dotted (green) line), and 0.4 (dashed (blue) line). In this figure, we show $\hat{y}$ as a (black) square and $\mu$ as a (black) dot.

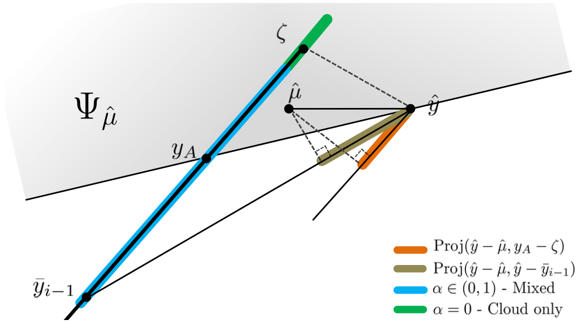
$\Psi_{\hat{\mu}}$
$\zeta$
$\hat{\mu}$
$\hat{y}$
$y_A$
$\bar{y}_{i-1}$
Proj$(\hat{y}-\hat{\mu}, y_A-\zeta)$
Proj$(\hat{y}-\hat{\mu}, \hat{y}-\bar{y}_{i-1})$
$\alpha \in (0,1)$ - Mixed
$\alpha = 0$ - Cloud only

Fig. 6. Illustrating first of the two possible configurations: $\zeta \in \Psi_{\hat{\mu}}$ (the closed half-plane defined by the line joining $\hat{y}$ and $y_A$ and contains the point $\hat{\mu}$) in the proof of Theorem 4.6.

Observe that in the regime given by (16), the angle between the vectors $y_A - \zeta$ and $\hat{y} - \hat{\mu}$ lies in the interval $[\pi/2, \pi]$. In this regime, for every $i \geq 1$, one out of the following two possibilities occurs.

i. $$\|\mathcal{P}(\hat{y}-\hat{\mu}, \hat{y}-\bar{y}_{i-1})\| \geq \|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\zeta)\|\,, \quad (19)$$

which holds whenever $\zeta$ is contained in $\Psi_{\hat{\mu}}$ [2]. This can be seen in Figure 6. Equality is achieved when $\bar{y}_{i-1}$ has a magnitude equal to infinity.

ii. $$\|\mathcal{P}(\hat{y}-\hat{\mu}, \hat{y}-\bar{y}_{i-1})\| \geq \|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\hat{y})\|\,, \quad (20)$$

which holds whenever $\zeta$ is not contained in the closure of the half plane defined by the line joining $\hat{y}$ and $y_A$ and which contains the point $\hat{\mu}$. This can be seen in Figure 7. Equality is achieved when $\bar{y}_{i-1}$ has a magnitude equal to infinity.

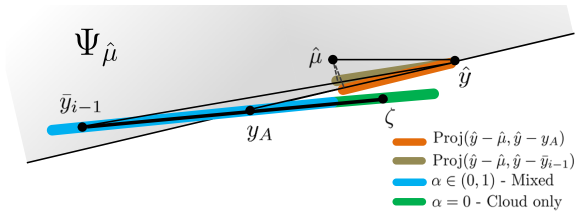
$\Psi_{\hat{\mu}}$
$\hat{\mu}$
$\hat{y}$
$\bar{y}_{i-1}$
$y_A$
$\zeta$
Proj$(\hat{y}-\hat{\mu}, \hat{y}-y_A)$
Proj$(\hat{y}-\hat{\mu}, \hat{y}-\bar{y}_{i-1})$
$\alpha \in (0,1)$ - Mixed
$\alpha = 0$ - Cloud only

Fig. 7. Illustrating the second of the two possible configurations: $\zeta \notin \Psi_{\hat{\mu}}$ (the closed half-plane defined by the line joining $\hat{y}$ and $y_A$ and contains the point $\hat{\mu}$) in the proof of Theorem 4.6.

Therefore, we conclude that

$$\begin{aligned}&\|\mathcal{P}(\hat{y}-\hat{\mu}, \hat{y}-\bar{y}_{i-1})\| \geq \\ &\min(\|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\zeta)\|, \|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\hat{y})\|)\,. \quad (21)\end{aligned}$$

Further, applying the triangle inequality to the set of points $\zeta$, $\hat{y}$, and $\bar{y}_{i-1}$, we obtain

$$\|\hat{y}-\bar{y}_{i-1}\| \leq \|\bar{y}_{i-1}-\zeta\| + \|\hat{y}-\zeta\|\,. \quad (22)$$

Combining (18), (21) and (22), we obtain

$$\|\bar{y}_i-\zeta\| \leq \frac{\|y_A-\zeta\|(\|\bar{y}_{i-1}-\zeta\| + \|\hat{y}-\zeta\|)}{\min(\|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\zeta)\|, \|\mathcal{P}(\hat{y}-\hat{\mu}, y_A-\hat{y})\|)}\,,$$

which converges if condition (17) is satisfied.

We show the necessity of (16) through the following construction. Suppose that condition (16) is violated. Equivalently, suppose that the angle between the vectors $y_A - \zeta$ and $\hat{y} - \hat{\mu}$ is in the interval $[0, \pi/2)$ (we refer the reader to Figure 3 for an illustration). Now, we can choose a $\bar{y}_0$ to be arbitrarily close to the red region ($\alpha = 1$). The coefficient in front of $\|y_A - \zeta\|$ in (18) can be made arbitrarily large. Therefore, in one iteration of the algorithm, $\alpha_1 = 1$. This establishes the necessity of (16). □

This result is numerically verified in Figure 8. In this figure, condition (16) is the vertical line passing through the point $\zeta$ (the black diamond). We can see that there are no green points that lie strictly to the right of this vertical line, thereby numerically verifying the necessity of (16).

Akin to the necessary condition, we seek a result which does not depend upon the knowledge of $\zeta$. Suppose that $\zeta \in \bar{\mathcal{C}}$, where $\bar{\mathcal{C}}$ is a set known to the sensor. Define the set $S_\zeta$ as the set of all points which satisfy the sufficient conditions from Theorem 4.6. Then, the following result holds.

[2] Recall that we defined $\Psi_{\hat{\mu}}$ to be the closed half-plane defined by the line joining $\hat{y}$ and $y_A$ and contains the point $\hat{\mu}$.

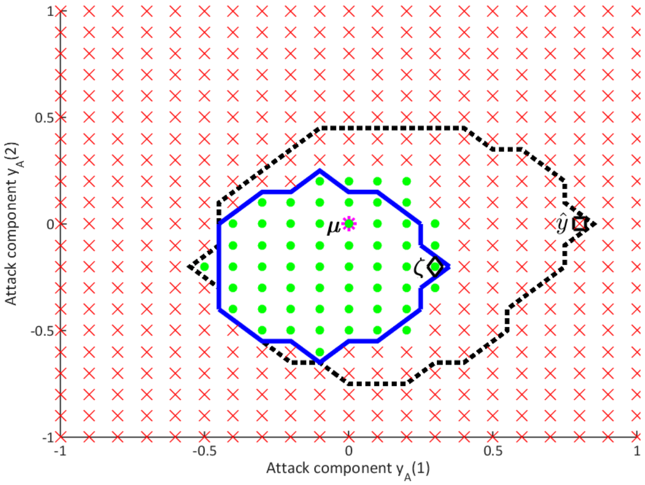

Fig. 8. Numerically generated plot to study the region of convergence for Algorithm 1 in two dimensions. In the planar case, the value of the mean $\hat{\mu} = \mu = (0, 0)$ (shown as a (magenta) star), the attacker's mean $\zeta = (0.3, -0.2)$ (shown as a (black) diamond) and the sensor's value $\hat{y} = (0.8, 0)$, shown as a (black) square. For every value of $y_A = [y_A(1), y_A(2)]$ on a grid in the neighborhood of $\mu$, we run Algorithm 1 for a set of different initial values $\bar{y}_0$. If the algorithm converges to an $\alpha^* \in (0, 1)$ for *every* choice of $\bar{y}_0$, then the corresponding point $y_A$ is shown as a (green) dot. Otherwise, it is shown as a (red) cross. The analytically derived sufficient condition from Theorem 4.6 is shown as a solid (blue) contour. The analytically derived necessary condition from Theorem 4.4 is shown as a dashed (black) contour.

**Corollary 4.7 (Sufficient Condition)** *A sufficient condition for strong convergence of Algorithm 1 is that* $y_A \in \bigcap_{\zeta \in \bar{\mathcal{C}}} S_\zeta$.

We now plot the analytic condition from Corollary 4.7 for different sets $\bar{\mathcal{C}}$. In particular, let the value of the mean $\hat{\mu} = \mu = (0, 0)$ (since this is a planar case), the sensor's value $\hat{y} = (0.8, 0)$, and the set $\bar{\mathcal{C}}$ be different circles of increasing radii around $\mu$. Then, Figure 9 shows how the set $\bigcap_{\zeta \in \bar{\mathcal{C}}} S_\zeta$ evolves with increasing radius of $\bar{\mathcal{C}}$. As is expected, the set computed for a smaller radius contains the set computed for one with higher radius. Since the sufficient condition is of interest to the attacker, we can see that with higher uncertainty in the mismatch between the means, the set of values of $y_A$ that guarantee convergence of Algorithm 1 shrinks in size, making it increasingly difficult for an attacker to always guarantee convergence of Algorithm 1.

## 5 Conclusion and Future Directions

This work introduced a novel approach to trusted computation when a central computer is leveraged but is likely to be compromised by an adversary. In our approach, we considered a sensor that may perform approximate but trusted computation on partial data, which is then fused with the output of the central computer in an optimal manner. We proposed a game-theoretic formulation and formalized an iterated best response algorithm. Formal statements were derived that characterize parameter regimes under which the iterative algorithm converges. The derived necessary and sufficient conditions become relatively tight in the case when the distributions of the unknown random variables used by the defender and the attacker to compute their respective cost functions have identical means. Numerical simulations validate our theoretical results.

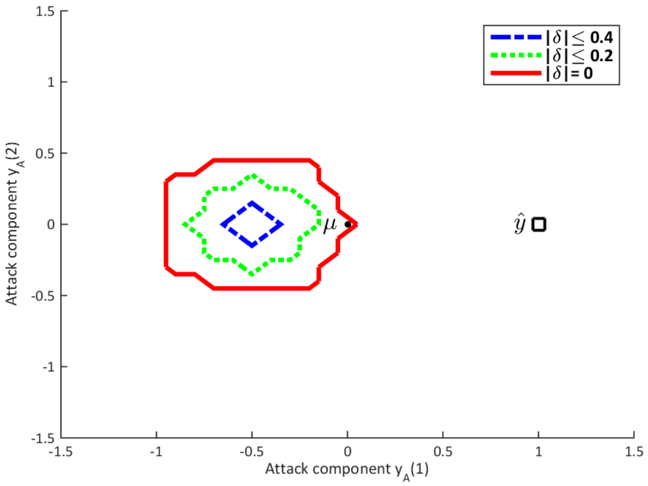

Fig. 9. Plot of how the analytic necessary condition from Corollary 4.7 evolves for increasing radii of the set $\bar{\mathcal{C}}$ which contains $\zeta$. This plot has been numerically generated by sampling 100 points uniformly randomly out of circles $\bar{\mathcal{C}}$ of radii equal to zero (solid (red) line), 0.2 (dotted (green) line), and 0.4 (dashed (blue) line). In this figure, we show $\hat{y}$ as a (black) square and $\mu$ as a (black) dot.

## References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. *Journal of the ACM*, 52(2), 2007.

[2] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against Support Vector Machines. In *Int'l Conf. on Machine Learning (ICML)*, 2012.

[3] S. D. Bopardikar, A. Speranzon, and C. Langbort. Trusted computation with an Adversarial Cloud. In *Proceedings of the American Control Conference*, pages 2445 – 2452, 2014.

[4] G. W. Brown. Iterative solutions of games by Fictitious Play. In *Activity Analysis of Production and Allocation, T. C. Koopmans, Ed. New York Wiley*, pages 374–376, 1951.

[5] P. Drineas and A. Zouzias. A note on element-wise matrix sparsification via a matrix-valued bernstein inequality. Arxiv report, 2011. Available online at `http://arxiv.org/pdf/1006.0407.pdf`.

[6] F. Farokhi, A. Texeira, and C. Langbort. Gaussian cheap talk game with quadratic cost functions: When herding between strategic senders is a virtue. In *Proceedings of the American Control Conference*, 2014.

[7] D. Fudenberg. *The theory of learning in games*, volume 2. MIT press, 1998.

[8] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM Workshop on Artificial Intelligence and Security (AISec)*, 2011.

[9] J. R. Marden, G. Arslan, and J. S. Shamma. Joint Strategy Fictitious Play with Inertia for Potential Games. *IEEE Transactions on Automatic Control*, 54(2):208–220, 2009.

[10] D. M. Reeves and M. P. Wellman. Computing best-response strategies in infinite games of incomplete information. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 470–478. AUAI Press, 2004.

[11] J. Robinson. An iterative method of solving a game. *Ann. Math.*, 54:296–301, 1951.

[12] C. Sawade, T. Scheffer, M. Br uckner, and T. Schffer. Bayesian games for adversarial regression problems. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

[13] J. S. Shamma and G. Arslan. Unified Convergence Proofs of Continuous-time Fictitious Play. *IEEE Transactions on Automatic Control*, 49(7):1137–1142, 2004.

[14] J. S. Shamma and G. Arslan. Dynamic Fictitious Play, Dynamic Gradient Play, and Distributed Convergence to Nash Equilibria. *IEEE Transactions on Automatic Control*, 50(3):312–327, 2005.