2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

On the security of randomly transformed quadratic programs for privacy-preserving cloud-based control

Philipp Binfet, Nils Schlüter, and Moritz Schulze Darup

Abstract— **Control related data, such as system states and inputs or controller specifications, is often sensitive. Meanwhile, the increasing connectivity of cloud-based or networked control results in vast amounts of such data, which poses a privacy threat, especially when evaluation on external platforms is considered. In this context, a cipher based on a random affine transformation gained attention, which is supposed to enable privacy-preserving evaluations of quadratic programs (QPs) with little computational overhead compared to other methods.**

This paper deals with the security of such randomly transformed QPs in the context of model predictive control (MPC). In particular, we show how to construct attacks against this cipher and thereby underpin concerns regarding its security in a practical setting. To this end, we exploit invariants under the transformations and common specifications of MPC-related QPs. Our numerical examples then illustrate that these two ingredients suffice to extract information from ciphertexts.

Index Terms— **Control Systems Privacy, Cyber-Physical Security, Quadratic Programming, Model Predictive Control**

I. Introduction

The privacy-preserving evaluation of a control related functionality is the main focus of encrypted control. In this context, quadratic programs (QPs) are of special interest, because they serve as the corner stone for solutions in many decision-making problems, where model predictive control (MPC) is a famous example. Roughly speaking, data privacy can be achieved via several cryptographic methods. Noteworthy are homomorphic encryption [1], [2], secure multi-party computation [3], and differential privacy [4]. However, these methods come with specific drawbacks such as large overhead in terms of computation or communication or a privacy-accuracy trade-off. Nonetheless, QPs have been addressed with these methods, for example in [5], [6].

A cipher which lately received attention is called random affine transformation (RT), random matrix encryption or affine masking. This method does not suffer from the aforementioned drawbacks and it can be used for a confidential evaluation of optimization problems, e.g., on a cloud. The application of RT ciphers to linear programming is addressed in [7]–[9], whereas QPs are considered in [10], [11]. A more general formulation for QPs can be found in [12] and [13], where the latter has a focus on linear MPC, as it is the case for [14]. Finally, also nonlinear MPC [15] and federated learning [16] have been considered in this context. Despite the fact that we collect these results under the umbrella of RT ciphers here, we point out that there are differences between them in terms of key reusage and additional permutations.

Philipp Binfet, Nils Schlüter, and Moritz Schulze Darup are with the Control and Cyber-physical Systems Group, Department of Mechanical Engineering, TU Dortmund University, Germany. E-mail correspondence to {philipp.binfet, nils.schlueter, moritz.schulzedarup}@tu-dortmund.de.

At this point, one may already suspect that the advantages of RTs do not come for free. In order to confirm that intuition, this paper deals with the construction of concrete attacks on the RTs applied to MPC, where QPs are solved sequentially. For the confidential evaluation of these QPs, we consider RTs over real numbers (as in the literature above) and allow for different keys in every problem instance. An extension with additional permutations is shown later. This way, many of the publications which use RTs are addressed. Now, as pointed out in [17], the "vector-ciphertexts" resulting from such an RT can be made secure. However, applying this cipher to QPs results in transformed QPs with slightly different "matrix-ciphertexts" for the parameters (such as the Hessian), which contain invariants. We show that certain information is inevitably leaked from ciphertexts. Furthermore, based on the invariants in combination with ciphertexts and little additional knowledge (justified by Kerckhoffs' principle) one can break the cipher entirely. Our findings are illustrated by an application to setpoint and tracking MPC problems.

In the remainder of the paper, we first specify the transformed/encrypted QP and the corresponding RT (Section II). Then, Section III analyzes the peculiarities of these QPs on which the attacks in Section IV are based.

II. Problem statement

We are interested in solving various instances of the QP

$$z_k^* := \arg\min_z \frac{1}{2} z^\top H_k z + f_k^\top z \quad \text{s.t.} \quad G_k z \leq e_k \tag{1}$$

where $k \in \mathbb{N}$ distinguishes the (time-) varying parameters

$$H_k \in \mathbb{R}^{l \times l}, \quad G_k \in \mathbb{R}^{q \times l}, \quad f_k \in \mathbb{R}^l, \quad \text{and} \quad e_k \in \mathbb{R}^q \tag{2}$$

with constant dimensions $l, q \in \mathbb{N}$. Further, we want to outsource this optimization to a cloud. Here, the (honest-but-curious) cloud represents any external computation platform that is interested in learning the QP parameters H_k, G_k, f_k, and e_k as well as the optimizer z_k^* but executes computations as specified. In order to establish privacy, it has been proposed in the literature (e.g., [13]) to use an RT of the optimization variable according to

$$z = R_k y + r_k, \tag{3}$$

where an invertible matrix $R_k \in \mathbb{R}^{l \times l}$ and a vector $r_k \in \mathbb{R}^l$ are randomly chosen for each k. With this at hand, one can transmit the transformed parameters

$$\tilde{H}_k := R_k^\top H_k R_k, \qquad \tilde{G}_k := G_k R_k, \tag{4a}$$

$$\tilde{f}_k := R_k^\top (f_k + H_k r_k), \qquad \tilde{e}_k := e_k - G_k r_k. \tag{4b}$$

to the cloud, which then solves

$$y_k^* := \arg\min_y \frac{1}{2} y^\top \tilde{H}_k y + \tilde{f}_k^\top y \quad \text{s.t.} \quad \tilde{G}_k y \leq \tilde{e}_k \tag{5}$$

instead of (1). Finally, based on the returned optimizer y_k^*, we easily recover the desired optimizer z_k^* via (3). Correctness of this scheme can be easily verified (cf. [11, Thm. 1]). However, the question of interest in this paper is whether the privacy of the original QP parameters (2) (and optimizers z_k^*) is indeed protected.

In this context, many authors conclude that (4) establishes privacy of (2) from the following observation. Assume the cloud came up with a guess $\hat{H}_k$, $\hat{G}_k$, $\hat{f}_k$, $\hat{e}_k$, $\hat{R}_k$, and $\hat{r}_k$ for the original QP parameters that is consistent with the transformations (4) and the data available to the cloud in terms of $\tilde{H}_k$, $\tilde{G}_k$, $\tilde{f}_k$, and $\tilde{e}_k$, i.e.,

$$\tilde{H}_k = \hat{R}_k^\top \hat{H}_k \hat{R}_k, \qquad \tilde{G}_k = \hat{G}_k \hat{R}_k, \tag{6a}$$

$$\tilde{f}_k = \hat{R}_k^\top (\hat{f}_k + \hat{H}_k \hat{r}_k), \qquad \tilde{e}_k = \hat{e}_k - \hat{G}_k \hat{r}_k. \tag{6b}$$

Then, the cloud could easily generate infinitely many additional consistent guesses $\check{H}_k$, $\check{G}_k$, $\check{f}_k$, $\check{e}_k$, $\check{R}_k$, and $\check{r}_k$ by choosing any invertible matrix $\tilde{R}_k \in \mathbb{R}^{l \times l}$ and any vector $\tilde{r}_k \in \mathbb{R}^l$ and by specifying the additional guesses via

$$\check{H}_k := \tilde{R}_k^\top \hat{H}_k \tilde{R}_k, \qquad \check{G}_k := \hat{G}_k \tilde{R}_k, \tag{7a}$$

$$\check{f}_k := \tilde{R}_k^\top (\hat{f}_k + \hat{H}_k \tilde{r}_k), \qquad \check{e}_k := \hat{e}_k - \hat{G}_k \tilde{r}_k, \tag{7b}$$

$$\check{R}_k := \tilde{R}_k^{-1} \hat{R}_k, \qquad \check{r}_k := \tilde{R}_k^{-1} (\hat{r}_k - \tilde{r}_k). \tag{7c}$$

Clearly, (7a) and (7b) just reflect another transformation of the form (4) using the chosen $\tilde{R}_k$ and $\tilde{r}_k$. The relations (7c) provide suitable updates of the guessed transformations. Now, (6) and (7) imply that even if one can solve the nonlinear system of equations underlying the transformations (4), there exist infinitely many other solutions, and it seems impossible to select the one that actually applies. In other words, privacy stems from ambiguity of the applied transformation. We will double-check this argumentation for popular problem specifications arising in model predictive control in the following.

A. Problem specifications for predictive control

Classical MPC leads to a sequence of optimal control problems in the form (1), where H_k and G_k are constant (see, e.g., [18, Eq. (7)]). This is summarized in the following specification for later reference.

Specification 1: The matrices H_k and G_k in (1) are constant, i.e., $H_k = H_0$ and $G_k = G_0$ for every $k \in \mathbb{N}$.

Furthermore, the optimization variable z typically represents the predicted input sequence in MPC, which is, among other constraints, often subjected to box constraints of the form

$$\underline{z} \leq z \leq \overline{z}. \tag{8}$$

Such or similar constraints lead to constant parts in e_k, as specified next.

Specification 2: At least the first $q_{\text{fix}} \in \mathbb{N}$ elements of the constraint vectors e_k are constant, i.e.,

$$e_k = \begin{pmatrix} e^{\text{fix}} \\ e_k^{\text{var}} \end{pmatrix} \quad \text{for every } k \in \mathbb{N} \tag{9}$$

with $e^{\text{fix}} \in \mathbb{R}^{q_{\text{fix}}}$ and $e_k^{\text{var}} \in \mathbb{R}^{q_{\text{var}}}$, where $q_{\text{var}} := q - q_{\text{fix}}$.

Finally, for classical MPC, the vectors e_k and f_k are affine in the current system state x_k [18, Eq. (7)]. Hence, e_k and f_k are (almost) constant whenever the states x_k are (almost) constant. The latter applies, for instance, if the state converges to a setpoint. Thus, we also consider the following specification, where we assume exactly constant vectors for analysis purposes.

Specification 3: The vectors f_k and e_k in (1) are constant for every k in some set $\mathcal{K} \subseteq \mathbb{N}$, i.e., $f_j = f_k$ and $e_j = e_k$ for all $j, k \in \mathcal{K}$.

We will investigate the implications of Specifications 1, 2, and 3 on the desired privacy in the upcoming sections. Before doing so, we present a variant of the transformations in (4) which can increase the security.

B. Additional random permutations

Inspired by [19, Sect. 3.2], we will also briefly discuss the combination of the transformations (4) with random permutations of the inequality constraints in (1). More precisely, for every $k \in \mathbb{N}$, a permutation matrix $P_k \in \{0,1\}^{q \times q}$ (in addition to R_k and r_k) is randomly chosen which results in

$$\tilde{G}_k' := P_k G_k R_k \quad \text{and} \quad \tilde{e}_k' := P_k (e_k - G_k r_k) \tag{10}$$

instead of the corresponding transformations in (4). Accordingly, we substitute the constraints $\tilde{G}_k y \leq \tilde{e}_k$ in (5) with $\tilde{G}_k' y \leq \tilde{e}_k'$. Note that, in light of such permutations, the assumed order of the constant and varying parts of e_k in (9) is without loss of generality.

III. Problem analysis

Before discussing possible attacks against the presented scheme, we analyze the transformed QP and the data available to the cloud. To avoid technicalities, we make the following assumption throughout the remaining paper.

Assumption 1: The matrices H_k are positive definite and the rank of G_k is l for every $k \in \mathbb{N}$.

A. Invariants and their relation to the dual problem(s)

We begin by pointing out two invariants regarding the transformed parameters (4). These are closely related to the dual of the QP (1), which is well-known to be

$$\lambda_k^* = \arg\min_{0 \leq \lambda} \frac{1}{2} \lambda^\top G_k H_k^{-1} G_k \lambda + (G_k H_k^{-1} f_k + e_k)^\top \lambda. \tag{11}$$

Interestingly, the duals of (1) and (5) are equal. In fact, this immediately follows from the invariants summarized next.

Lemma 1: The transformed parameters (4) are related to the original parameters via

$$\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top = G_k H_k^{-1} G_k^\top \qquad \text{and} \tag{12a}$$

$$\tilde{G}_k \tilde{H}_k^{-1} \tilde{f}_k + \tilde{e}_k = G_k H_k^{-1} f_k + e_k \tag{12b}$$

for any choice of R_k and r_k.

Proof: The invariants immediately follow from substituting the expressions (4) for $\tilde{H}_k$, $\tilde{G}_k$, $\tilde{f}_k$, and $\tilde{e}_k$. ∎

Moreover, the invariants (12) are closely related to Specifications 1 and 3 as detailed next.

Lemma 2: If Specification 1 applies, then $\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top$ is constant for every $k \in \mathbb{N}$. Further, if Specifications 1 and 3 apply, then $\tilde{G}_k \tilde{H}_k^{-1} \tilde{f}_k + \tilde{e}_k$ is constant for every $k \in \mathcal{K}$.

Proof: Specification 1 obviously implies that the right-hand side in (12a) is constant for every $k \in \mathbb{N}$. Due to the invariant, this property directly translates to $\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top$. The proof for $\tilde{G}_k \tilde{H}_k^{-1} \tilde{f}_k + \tilde{e}_k$ being constant for every $k \in \mathcal{K}$ under Specifications 1 and 3 is analogous. ∎

Based on Lemma 2, the cloud obtains necessary conditions for checking whether certain specifications apply. Before exploiting this feature in Section IV, we briefly note that invariants similar to (12) also arise if the permuted parameters $\tilde{G}'_k$ and $\tilde{e}'_k$ (as specified in (10)) replace $\tilde{G}_k$ and $\tilde{e}_k$ in (4). In fact, we then find

$$\tilde{G}'_k \tilde{H}_k^{-1} (\tilde{G}'_k)^\top = P_k G_k H_k^{-1} G_k^\top P_k^\top \qquad \text{and} \tag{13a}$$

$$\tilde{G}'_k \tilde{H}_k^{-1} \tilde{f}_k + \tilde{e}'_k = P_k \left(G_k H_k^{-1} f_k + e_k \right). \tag{13b}$$

Finally, also the set of active constraints (or, if permutations are involved, the set's cardinality) is invariant under (4).

B. Consistent and correct guesses

Next, we focus on the systematic derivation of guesses (for $\hat{H}_k$, $\hat{G}_k$, $\hat{f}_k$, $\hat{e}_k$, $\hat{R}_k$, and $\hat{r}_k$) that are consistent with the observed data according to (6). It is easy to see that a trivial solution to this task is

$$\hat{H}_k := \tilde{H}_k, \qquad \hat{G}_k := \tilde{G}_k, \qquad \hat{R}_k := I_l, \tag{14a}$$

$$\hat{f}_k := \tilde{f}_k, \qquad \hat{e}_k := \tilde{e}_k, \qquad \hat{r}_k := 0. \tag{14b}$$

While trivial, the family of associated guesses (7) includes the original parameters as apparent from the (existing but typically unknown) transformation with $\tilde{R}_k := R_k^{-1}$ and $\tilde{r}_k = -\tilde{R}_k r_k$, which indeed leads to the correct guess

$$\check{H}_k = H_k, \qquad \check{G}_k = G_k, \qquad \check{R}_k = R_k, \tag{15a}$$

$$\check{f}_k = f_k, \qquad \check{e}_k = e_k, \qquad \check{r}_k = r_k. \tag{15b}$$

Remarkably, any consistent guess can, in principle, be transformed into a correct guess via (7) as formalized in the following lemma.

Lemma 3: Assume a consistent guess satisfying (6) is known. Then, the transformation (7) with $\tilde{R}_k := \hat{R}_k R_k^{-1}$ and $\tilde{r}_k := \hat{r}_k - \tilde{R}_k r_k$ yields a correct guess satisfying (15).

Proof: We initially note that $\check{R}_k = R_k$ and $\check{r}_k = r_k$ immediately results when substituting $\tilde{R}_k$ and $\tilde{r}_k$ in (7c). Further, substituting $\tilde{R}_k$ in (7a) and using (6a) and (4a) proves $\check{H}_k = H_k$ and $\check{G}_k = G_k$. Analogously, we can prove $\check{f}_k = f_k$ and $\check{e}_k = e_k$ by exploiting (4b) and (6b). ∎

While consistent with the data, the trivial guess (14) is not very helpful if Specification 1 applies because, up until this point, consistency has only been defined with respect to isolated instances k (typically leading to guesses that are not consistent across different instances, e.g., $\hat{H}_0 \neq \hat{H}_k$). In order to avoid this issue, we derive another consistent guess based on the invariant (12a), which promotes cross-instance consistency according to Lemma 2. This novel guess builds on a singular value decomposition (SVD) of the matrix on the left-hand side of (12a) leading to

$$\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top, \tag{16}$$

where $\tilde{U}_k := \begin{pmatrix} \tilde{u}_{1,k} & \dots & \tilde{u}_{q,k} \end{pmatrix} \in \mathbb{R}^{q \times q}$ and $\tilde{V}_k$ are orthogonal and where

$$\tilde{\Sigma}_k = \begin{pmatrix} \tilde{D}_k & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{q \times q} \text{ with } \tilde{D}_k := \operatorname{diag}(\tilde{\sigma}_{1,k}, \dots, \tilde{\sigma}_{l,k}).$$

The singular values $\tilde{\sigma}_{1,k} \geq \dots \geq \tilde{\sigma}_{l,k} > 0$ are l in number, since Assumption 1 immediately implies that $G_k H_k^{-1} G_k^\top$ is of rank l. Due to (12a), the same applies to $\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top$ yielding l positive $\tilde{\sigma}_{i,k}$. We then find the following theorem.

Theorem 4: The guess with the parameters

$$\hat{H}_k := \tilde{D}_k^{-1}, \quad \hat{G}_k := \begin{pmatrix} \tilde{u}_{1,k} & \dots & \tilde{u}_{l,k} \end{pmatrix}, \text{ and } \hat{R}_k := \hat{G}_k^\top \tilde{G}_k$$

is consistent with (6a).

Proof: To prove the claim, we first note that $\hat{G}_k^\top \tilde{U}_k = \begin{pmatrix} I_l & 0 \end{pmatrix}$ by construction of $\hat{G}_k$. Furthermore, since $\tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top$ is positive semi-definite with rank l, the first l columns of $\tilde{V}_k$ are equivalent to those of $\tilde{U}_k$. Hence, $\hat{G}_k^\top \tilde{V}_k = \begin{pmatrix} I_l & * \end{pmatrix}$. Next, we multiply (16) with $\hat{G}_k^\top$ from the left and with $\hat{G}_k$ from the right to obtain

$$\hat{R}_k \tilde{H}_k^{-1} \hat{R}_k^\top = \begin{pmatrix} I_l & 0 \end{pmatrix} \tilde{\Sigma}_k \begin{pmatrix} I_l \\ * \end{pmatrix} = \tilde{D}_k.$$

Since only square matrices are involved and $\tilde{D}_k$ is regular by construction, this relation implies invertibility of $\hat{R}_k$ and $\tilde{D}_k^{-1} = \hat{R}_k^{-\top} \tilde{H}_k \hat{R}_k^{-1}$. With this at hand, we find

$$\hat{R}_k^\top \hat{H}_k \hat{R}_k = \hat{R}_k^\top \tilde{D}_k^{-1} \hat{R}_k = \tilde{H}_k,$$

which proves the first equation in (6a). It remains to prove $\tilde{G}_k = \hat{G}_k \hat{R}_k$. Hence, we define $\tilde{Y}_k := \begin{pmatrix} \tilde{u}_{l+1,k} & \dots & \tilde{u}_{q,k} \end{pmatrix}$ and note that $\tilde{Y}_k^\top \tilde{U}_k = \begin{pmatrix} 0 & I_{q-l} \end{pmatrix}$ and $\tilde{Y}_k^\top \tilde{V}_k = \begin{pmatrix} 0 & * \end{pmatrix}$. As a consequence, we find $\tilde{Y}_k^\top \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top \tilde{Y}_k = 0$. Due to (16), this also implies $\tilde{Y}_k^\top \tilde{G}_k \tilde{H}_k^{-1} \tilde{G}_k^\top \tilde{Y}_k = 0$. Now, due to positive definiteness of $\tilde{H}_k^{-1}$, the latter relation holds if and only if $\tilde{Y}_k^\top \tilde{G}_k = 0$. Furthermore, we have $\hat{G}_k \hat{G}_k^\top = I_q - \tilde{Y}_k \tilde{Y}_k^\top$ due to $\tilde{U}_k \tilde{U}_k^\top = I_q$. Hence, we obtain

$$\hat{G}_k \hat{R}_k = \hat{G}_k \hat{G}_k^\top \tilde{G}_k = (I_q - \tilde{Y}_k \tilde{Y}_k^\top) \tilde{G}_k = \tilde{G}_k,$$

which completes the proof. ∎

Theorem 4 indicates that finding consistent guesses according to (6) can be decoupled. In fact, consistent $\hat{H}_k$, $\hat{G}_k$, and $\hat{R}_k$ can be identified by only considering (6a) and $\hat{f}_k$, $\hat{e}_k$, and $\hat{r}_k$ only appear in (6b). In particular, once consistent $\hat{H}_k$, $\hat{G}_k$, and $\hat{R}_k$ have been found, (6b) can be rewritten in terms of the linear equations

$$\begin{pmatrix} I_q & 0 & -\hat{G}_k \\ 0 & \hat{R}_k^\top & \hat{R}_k^\top \hat{H}_k \end{pmatrix} \begin{pmatrix} \hat{e}_k \\ \hat{f}_k \\ \hat{r}_k \end{pmatrix} = \begin{pmatrix} \tilde{e}_k \\ \tilde{f}_k \end{pmatrix} \tag{17}$$

with the unknowns $\hat{f}_k$, $\hat{e}_k$, and $\hat{r}_k$. These equations are, e.g., solved by $\hat{f}_k := \hat{R}_k^{-\top} \tilde{f}_k$, $\hat{e}_k := \tilde{e}_k$, and $\hat{r}_k := 0$. However, having $q+l$ equations for $q+2l$ unknowns, it is immediately clear that the system of equations is underdetermined and (infinitely many) more solutions exist. Moreover, the particular solution mentioned above is typically inconsistent with the Specifications 2 and 3. As a consequence, we investigate (17) in more detail in the next section.

C. Exploiting linear dependencies of parameters

Similar to Specification 1, also Specifications 2 and 3 call for consistent guesses across multiple problem instances. In the following, we will analyze the effect of these specifications on (17), assuming throughout that Specification 1 applies as well. For ease of presentation, we begin by analyzing the effect of Specification 3. In this context, we assume that the index set $\mathcal{K}$ contains s elements and enumerate them by $k_1, \dots, k_s$. Then, taking $\hat{e}_{k_1} = \dots = \hat{e}_{k_s}$ and $\hat{f}_{k_1} = \dots = \hat{f}_{k_s}$ into account, (17) results in the following relation:

$$\begin{pmatrix} I_q & 0 & -\hat{G}_0 & & 0 \\ \vdots & \vdots & & \ddots & \\ I_q & 0 & 0 & & -\hat{G}_0 \\ 0 & \hat{R}_{k_1}^\top & \hat{R}_{k_1}^\top \hat{H}_0 & & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & \hat{R}_{k_s}^\top & 0 & & \hat{R}_{k_s}^\top \hat{H}_0 \end{pmatrix} \begin{pmatrix} \hat{e}_{k_1} \\ \hat{f}_{k_1} \\ \hat{r}_{k_1} \\ \vdots \\ \hat{r}_{k_s} \end{pmatrix} = \begin{pmatrix} \tilde{e}_{k_1} \\ \vdots \\ \tilde{e}_{k_s} \\ \tilde{f}_{k_1} \\ \vdots \\ \tilde{f}_{k_s} \end{pmatrix}. \quad (18)$$

At this point, we note that based on $\hat{R}_{k_1}$, $\hat{H}_0$, and $\hat{G}_0$ (e.g., obtained via Theorem 4) one can construct the coefficient matrix in (18) because the remaining keys $\hat{R}_{k_2}, \hat{R}_{k_3}, \ldots, \hat{R}_{k_s}$ follow from $\hat{G}_0$ and $\tilde{G}_k$ with $k \in \mathcal{K}$. Now, at first glance, this system of equations looks overdetermined with $(q+l)s$ equations and $q+l(s+1)$ unknowns. However, it is easy to show (but omitted here for brevity) that the rank of the matrix in (18) is only $q+ls$. Hence, the system of equations (18) is again underdetermined and l (independent) equations are again missing. In contrast to (17), however, where l equations were missing for *each* $k \in \mathbb{N}$, here l equations are missing to identify all unknowns for *all* instances $k \in \mathcal{K}$.

In addition, we can exploit that the simultaneous application of Specifications 1 and 3 also implies identical QPs (1) for $k \in \mathcal{K}$ and consequently $z_{k_1}^* = \cdots = z_{k_s}^*$. We can make use of this relationship by noting that, once a guess $\hat{z}_{k_i}^*$ for any original optimizers $z_{k_i}^*$ is known, we can immediately derive $\hat{r}_{k_i} := \hat{z}_{k_i}^* - \hat{R}_{k_i} y_{k_i}^*$, $\hat{e}_{k_i} := \tilde{e}_{k_i} + \hat{G}_0 \hat{r}_{k_i}$, and $\hat{f}_{k_i} := \hat{R}_k^{-\top} \tilde{f}_{k_i} - \hat{H}_0 \hat{r}_{k_i}$.

It remains to comment on the effects of Specification 2. Although this specification addresses all $k \in \mathbb{N}$, we consider the same set $\mathcal{K}$ as above for convenience. In this context, (17) leads to a system of linear equations similar to (18) and also with the same right-hand side but with the unknown vectors

$$\hat{e}^{\text{fix}}, \hat{e}_{k_1}^{\text{var}}, \ldots, \hat{e}_{k_s}^{\text{var}}, \hat{f}_{k_1}, \ldots, \hat{f}_{k_s}, \hat{r}_{k_1}, \ldots, \hat{r}_{k_s}. \quad (19)$$

Hence, we (initially) have $q_{\text{fix}} + (q_{\text{var}} + 2l)s$ unknowns here but again $(q+l)s$ equations. Now, assuming

$$q_{\text{fix}} > l \qquad \text{and} \qquad s \geq \frac{q_{\text{fix}}}{q_{\text{fix}} - l}, \quad (20)$$

the number of equations exceeds (or matches) the number of unknowns. However, even if (20) holds, one can show that the rank deficiency of the coefficient matrix characterizing the system of linear equations is again l. Still, once we have a guess for either $\hat{e}^{\text{fix}}$ or any of the $\hat{f}_{k_i}$ or $\hat{r}_{k_i}$, we can compute all unknowns (19). We make use of this feature in Section IV.

D. Partially resolving permutations

As apparent from (13), the invariants (12) exploited above lose some of their informative value if permutations are involved. Nevertheless, if Specification 1 applies, the entries of $\tilde{G}_k' \tilde{H}_k^{-1} (\tilde{G}_k')^\top$ and $\tilde{G}_0' \tilde{H}_0^{-1} (\tilde{G}_0')^\top$ differ only in their position. Thus, whenever these matrices contain at least q distinct entries, which is often the case, it is straightforward to identify relative permutations $\Delta P_{k,0} := P_k P_0^\top$ such that

$$\tilde{G}_k' \tilde{H}_k^{-1} (\tilde{G}_k')^\top = \Delta P_{k,0} \tilde{G}_0' \tilde{H}_0^{-1} (\tilde{G}_0')^\top \Delta P_{k,0}^\top.$$

Moreover, if also Specification 3 applies, the identification can even be simplified based on relations like

$$\tilde{G}_k' \tilde{H}_k^{-1} \tilde{f}_k + \tilde{e}_k' = \Delta P_{k,0} \big(\tilde{G}_0' \tilde{H}_0^{-1} \tilde{f}_0 + \tilde{e}_0' \big).$$

Now, suppose that P_0 (or any other "absolute" permutation) is known. Then one can resolve all other permutations by evaluating $P_k = \Delta P_{k,0} P_0$. This eventually allows computing $\tilde{G}_k = P_k^\top \tilde{G}_k'$ and $\tilde{e}_k = P_k^\top \tilde{e}_k'$ such that we deal with the unpermuted problem again.

IV. Attack scenarios

In this section, we consider an exemplary application of the transformed QP (5) in the context of privacy-preserving MPC and illustrate how our theoretical findings about the transformation itself and the transformed parameters (4) can be used to launch attacks. The control scheme will be applied to a mobile robot, for which we briefly clarify the corresponding system model, the MPC setup, and the cloud-based solution of the resulting QP next.

A. Specification and cloud-based realization of MPC

We assume linear discrete-time system dynamics modeled by $x(k+1) = Ax(k) + Bu(k)$ with

$$A := I_2 \otimes \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad B := I_2 \otimes \begin{pmatrix} 0.5 \\ 1 \end{pmatrix},$$

where "$\otimes$" denotes the Kronecker product. The system is subject to the state and input constraints $\underline{x} \leq x(k) \leq \overline{x}$ and $\underline{u} \leq u(k) \leq \overline{u}$, respectively, with

$$\overline{x} = -\underline{x} := \begin{pmatrix} 20 & 5 & 20 & 5 \end{pmatrix}^\top \text{ and } \overline{u} = -\underline{u} := \begin{pmatrix} 1 & 1 \end{pmatrix}^\top. \quad (21)$$

In each time-step k, we minimize the cost function

$$\sum_{\kappa=k}^{k+N-1} \|y(\kappa) - y_{\text{ref}}(\kappa)\|_Q^2 + \|u(\kappa) - u(\kappa-1)\|_R^2$$

subject to the dynamics and constraints from above for $N = 5$ prediction steps, where the outputs $y(k) := (x_1(k) \;\; x_3(k))^\top$ reflect the robot's position. The weighting matrices are chosen as $Q = I_2$ and $R = 0.1 I_2$. As for the references, we will first consider $y_{\text{ref}}(k) = 0$ reflecting a setpoint and then a sinusoidal reference representing a circle with radius 10 traversed in counterclockwise direction with a period of 20 time steps. For both numerical experiments, the robot's initial state is $x(0) = (10 \;\; -2 \;\; 10 \;\; 2)^\top$ and $u(-1) = 0$. Now, reformulating the control task in terms of the QP (1) is straightforward, and we omit details for brevity. We just note that (after condensation) the decision variables z reflect the predicted input sequences in terms of $u(k), \ldots, u(k+N-1)$. Furthermore, for every $k \in \mathbb{N}$, the QP parameters (2) can be stated as

$$H_k = H_0, \;\; G_k = G_0 := \begin{pmatrix} I_l \\ -I_l \\ * \end{pmatrix}, \;\; e_k := \begin{pmatrix} \overline{z} \\ -\underline{z} \\ \varepsilon(x(k)) \end{pmatrix}, \quad (22a)$$

$$f_k := \varphi(x(k), u(k-1), y_{\text{ref}}(k), \ldots, y_{\text{ref}}(k+N-1)) \quad (22b)$$

with $\varepsilon : \mathbb{R}^n \to \mathbb{R}^{2nN}$ and $\varphi := \mathbb{R}^{n+m+Np} \to \mathbb{R}^l$, where $m = 2$, $n = 4$, and $p = 2$ reflect the input, state, and output dimension of the system at hand and where the number of decision variables is $l = mN = 10$. The constraint vectors $\underline{z}$ and $\overline{z}$ result from stacking the input constraints in (21). Finally, in each time step, we apply the optimal input $u^*(k)$ (the first m entries of z_k^*) to the system and

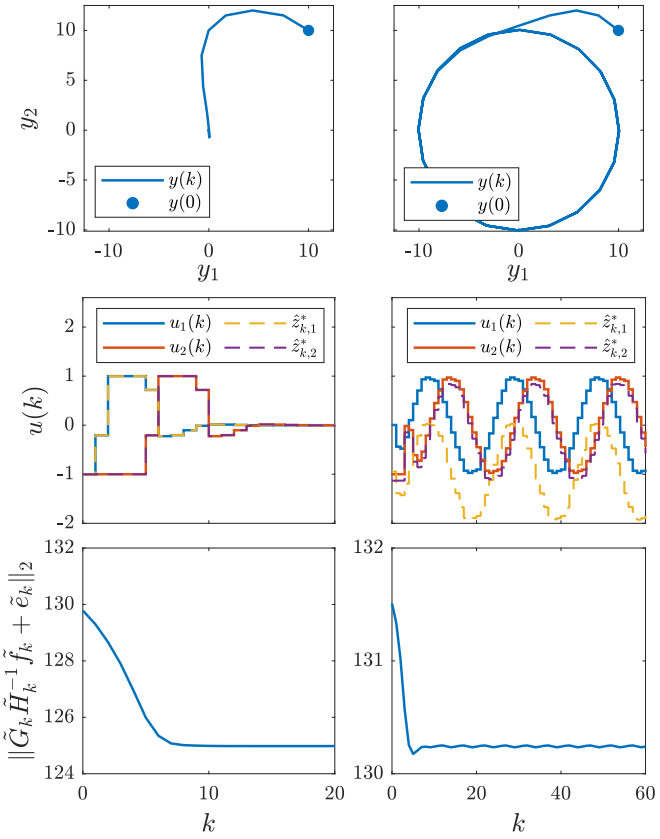

Fig. 1. Output trajectories (top), input sequences (middle), and norms of (12b) (bottom) for the first (left, setpoint) and second (right, tracking) experiment, respectively.

repeat the procedure at the next sampling instant $k+1$. For an overview, the resulting system trajectories and inputs for both references are illustrated in the upper and middle charts of Figure 1, respectively.

Finally, a cloud-based solution of the QPs via (5) is realized as follows. First, we randomly and independently choose R_k and r_k for each k. Since suitable distributions are typically not specified in the corresponding literature, we sample floating point numbers uniformly from the interval $[-10, 10]$ for simplicity here. Then, the cloud receives the transformed parameters $\tilde{H}_k$, $\tilde{G}_k$, $\tilde{e}_k$, and $\tilde{f}_k$ (computed according to (4)), solves (5), and returns y_k^* to the client, where z_k^* is recovered via (3).

B. Preparing and launching attacks

Summarizing Section III, we rely mostly on the relations between the transformed and original parameters stemming from Specifications 1 to 3 here. Via Lemma 2, we can see that constant $\tilde{G}_k\tilde{H}_k^{-1}\tilde{G}_k^\top$ (for all $k \in \mathbb{N}$) and $\tilde{G}_k\tilde{H}_k^{-1}\tilde{f}_k + \tilde{e}_k$ (for all k in some set $\mathcal{K} \subseteq \mathbb{N}$) are strong indicators for the validity of Specification 1 and 3, respectively. Here (due to (22a)), the cloud will observe constant $\tilde{G}_k\tilde{H}_k^{-1}\tilde{G}_k^\top$ for all k in both experiments and will thus (correctly) assume that Specification 1 applies. Regarding the second invariant, we depict $\|\tilde{G}_k\tilde{H}_k^{-1}\tilde{f}_k + \tilde{e}_k\|_2$ in the bottom charts of Figure 1. Remarkably, because f_k and e_k depend on the current state, some properties of the system's behavior are leaked inevitably. For instance, the convergence time to the reference and, in case of the tracking problem, the fact that the motion is periodic are revealed. Moreover, $\|\tilde{G}_k\tilde{H}_k^{-1}\tilde{f}_k + \tilde{e}_k\|_2$ is (almost) constant for all $k \in \mathcal{K}_1 := \{8, 9, \dots, 20\}$ in the first experiment and for various triplets such as $\{15, 20, 25\}$ or $\mathcal{K}_2 := \{10, 30, 50\}$ in the second one. A closer investigation of the involved vectors reveals that in the first experiment also $\tilde{G}_k\tilde{H}_k^{-1}\tilde{f}_k + \tilde{e}_k$ is constant for all $k \in \mathcal{K}_1$, and the cloud will thus (correctly) assume that Specification 3 applies at these time steps. Regarding the second experiment, the situation is slightly more complicated. In fact, while the norm is (almost) constant for both of the above triplets, the vectors $\tilde{G}_k\tilde{H}_k^{-1}\tilde{f}_k + \tilde{e}_k$ vary for the first triplet but are (almost) constant for $k \in \mathcal{K}_2$. Based on these observations, the cloud not only gets an indication for the validity of Specification 3 during some time steps for the second experiment, but it can even infer the (correct) reference period of 20 time steps.

Now, in order to launch an actual attack aiming for a reconstruction of the original QP parameters, the cloud could make use of Theorem 4 to derive consistent guesses for H_0, G_0, and all R_k. However, due to the special but common structure of G_0 from (22a), the cloud can obtain a more powerful guess more easily. In fact, it will receive the transformed matrices $\tilde{G}_k^\top = \begin{pmatrix} R_k^\top & -R_k^\top & * \end{pmatrix}$ and observe that the second square block is always a negation of the first one. Having already a strong indication for a constant $G_k = G_0$ and likely knowing about the common structure in MPC, it will simply pick the first square block of $\tilde{G}_k$ as the (correct) guess $\hat{R}_k$. Immediately, this leads to the (likewise correct) guesses $\hat{G}_0 = \tilde{G}_k\hat{R}_k^{-1}$ and $\hat{H}_0 = \hat{R}_k^{-\top}\tilde{H}_k\hat{R}_k^{-1}$.

In order to completely break the cipher, it remains to reconstruct e_k, f_k, and r_k. Hence, the cloud builds up the system of equations (18) for the steps in $\mathcal{K}_i$ with $i \in \{1, 2\}$. Now, as pointed out in Section III-C, it is necessary to add l additional equations to avoid that the system is underdetermined. As proposed, the cloud will add a guess for the constant z_k^* during the instances in $\mathcal{K}_i$, respectively. For the first experiment, the lower left chart in Figure 1 suggests that the system converges to an equilibrium. Hence, $\hat{z}_k^* = 0$ is a reasonable guess here. In the absence of more reasonable alternatives, the cloud will use the same guess for the second example, although it is most likely erroneous there. After adding $\hat{z}_k^* = 0$, the cloud solves (18) and obtains $\hat{e}_k$, $\hat{f}_k$, and $\hat{r}_k$ for all $k \in \mathcal{K}_i$.

Finally, to also address the steps not contained in the sets $\mathcal{K}_i$, the cloud (correctly) assumes that Specification 2 holds, which is reasonable based on the assumed structure of $\hat{G}_0$. In fact, the blocks I_l and $-I_l$ typically refer to constant box constraints as in (8). This corresponds to $q_{\text{fix}} = 2l$ implying that the first condition in (20) is satisfied and that the second holds whenever $s \geq 2$. Hence, the cloud straightforwardly includes all remaining time steps k in the system of equations for the unknowns (19) and adds an already solved one from $\mathcal{K}_i$ to ensure that the system is determined. After solving for these unknowns, all $\hat{z}_k^*$ can be computed via (3). The first two components of $\hat{z}_k^*$ are illustrated in the middle charts of Figure 1. As apparent, the inputs applied to the system are accurately reconstructed for

the first experiment. In the second experiment, despite the erroneous guess $\hat{z}_k^* = 0$ for $k \in \mathcal{K}_2$, only a constant offset is present in the reconstructed input sequences (the shape of the signals is accurately recovered). Note that the magnitude of the error in each component varies with the choice of $\hat{z}_k^*$. Since $z_{k,2}^* \approx 0 = \hat{z}_{k,2}^*$ for all $k \in \mathcal{K}_2$, the second component of the inputs is almost correctly reconstructed here.

C. Scenarios with permutations

We briefly address the case, where the transformations are accompanied by permutations according to (10). While this simple modification significantly complicates a reconstruction, we pointed out in Section III-D that relative permutations such as $\Delta P_{k,0}$ can often be identified with moderate effort. This is indeed the case in our experiments, even though the special structure of G_0 (together with the choices for Q and R) leads to ambiguous entries in $\tilde{G}'_k \tilde{H}_k^{-1} (\tilde{G}'_k)^\top$. Hence, picking up the assumption from Section III-D that P_0 is known, we can resolve all permutations for the experiments at hand and subsequently handle the reconstruction of the QP parameters as before. Regarding the identification of an unknown P_0, we note that the special structure of G_0 excludes many realizations of P_0. However, without additional knowledge, the problem of uniquely identifying P_0 is intractable.

D. Remarks on the attacks and additional knowledge

We conclude this section with some remarks on the proposed attacks. As apparent from Figure 1, the cloud is, in principle, able to infer crucial characteristics of the control system and to recover the majority of the sensitive data. However, the cloud is currently not able to verify its guesses without additional knowledge. In fact, this corresponds to a ciphertext-only setup which is assumed in the literature. Still, having suitable additional knowledge (e.g., exactly knowing that G_k or e_k have the structure in (22a)) seems reasonable for many real-world applications. Further, it is in line with Kerckhoffs' principle [20, Sect. 1.2], which (translated to this setup) states that the security of a cipher should not depend on an attackers' knowledge about the control system. Especially, knowledge about the plant, which has been excluded here, is well-suited to validate the guesses from Section IV-B resulting in a more targeted attack.

Furthermore, the most important observation resulting from our analysis is that only little information, which might even be public or easy to obtain, can enable to completely break RT ciphers in the context of QPs. Remarkably, this major issue is not present in other cryptosystems (such as homomorphic encryption [1], [2]), which could be used to ensure a privacy-preserving evaluation of QPs at the price of a higher computational load.

V. Conclusions and Outlook

This paper deals with the security of random affine transformations in the context of model predictive control, where it is used for the private evaluation of quadratic programs. We show that the arising ciphertexts still contain information that can be exploited for an attack and, most importantly, that little additional information can suffice to break all ciphertexts even though keys are not reused in our setup.

In future work, additional equality constraints (as in [10]) and a more detailed treatment of permutations are of interest. In this context, system knowledge, closed-loop effects, and the implementation over floating point numbers (see [17]), entail valuable angles of attacks that we neglected here.

Acknowledgment

Financial support by the German Research Foundation and the Daimler and Benz Foundation under the grants SCHU 2940/4-1 and 32-08/19 is gratefully acknowledged.

References

[1] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.

[2] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Intl. Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 409–437.

[3] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure multiparty computation and secret sharing*. Cambridge University Press, 2015.

[4] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[5] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based quadratic optimization with partially homomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2357–2364, 2020.

[6] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 1, pp. 395–408, 2016.

[7] J. Vaidya, "Privacy-preserving linear programming," in *Proc. of the ACM symposium on Applied Computing*, 2009, pp. 2002–2007.

[8] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," in *Proc. of the Int. Conf. on Privacy, Security, Risk and Trust*, 2011, pp. 916–924.

[9] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proc. of the IEEE Infocom*, 2011, pp. 820–828.

[10] L. Zhou and C. Li, "Outsourcing large-scale quadratic programming to a public cloud," *IEEE Access*, vol. 3, pp. 2581–2589, 2015.

[11] Z. Xu and Q. Zhu, "Secure and Resilient Control Design for Cloud Enabled Networked Control Systems," in *Proc. of the ACM Workshop on Cyber-Physical Systems-Security and/or Privacy*. Association for Computing Machinery, 2015, p. 31–42.

[12] P. C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras, "Perse privacy preserving solution methods based on optimization," in *Proc. of the Conference on Decision and Control*, 2013, pp. 206–211.

[13] A. Sultangazin and P. Tabuada, "Symmetries and isomorphisms for privacy in control over the cloud," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 538–549, 2020.

[14] A. M. Naseri, W. Lucia, and A. Youssef, "A Privacy Preserving Solution for Cloud-Enabled Set-Theoretic Model Predictive Control," in *Proc. of the European Control Conference*, 2022, pp. 894–899.

[15] K. Zhang, Z. Li, Y. Wang, and N. Li, "Privacy-preserved nonlinear cloud-based model predictive control via affine masking," *arXiv preprint arXiv:2112.10625*, 2021.

[16] H. Hayati, C. Murguia, and N. van de Wouw, "Privacy-preserving federated learning via system immersion and random matrix encryption," *arXiv preprint arXiv:2204.02497*, 2022.

[17] N. Schlüter, P. Binfet, and M. Schulze Darup, "Cryptanalysis of Random Affine Transformations for Encrypted Control," in *Proc. of the IFAC World Congress*, 2023, pp. 12 031–12 038.

[18] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[19] S. Salinas, C. Luo, W. Liao, and P. Li, "Efficient secure outsourcing of large-scale quadratic programs," in *Proc. of the ACM on Asia Conf. on Computer and Communications Security*, 2016, pp. 281–292.

[20] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.