Towards Encrypted MPC for Linear Constrained Systems

Moritz Schulze Darup, Adrian Redder, Iman Shames, Farhad Farokhi, and Daniel Quevedo

Abstract—We present an encrypted model predictive control (MPC) scheme for linear constrained systems. More precisely, we show that homomorphic encryption can be used to realize a secure and private cloud-based evaluation of an MPC if the corresponding piecewise affine control law is explicitly given.

Index Terms—Networked control systems, security, privacy, homomorphic encryption, predictive control for linear systems.

I. INTRODUCTION

THE INTERNET of Things (IoT), cyber-physical systems, and smart grids show that the requirements for control engineering are changing. Already today, a variety of control schemes are implemented on network or cloud services. While these architectures offer many opportunities, they also increase the risk of cyberattacks. Hence, much research has focused on safe implementations of control schemes within clouds (see [1], [2]). In this context, homomorphic encryption (HE) seems to be a promising tool (see [3], [4]). HE allows simple arithmetic operations (e.g., sums or multiplications) to be evaluated on ciphertext. It thus opens the opportunity to implement simple control schemes (e.g., linear state feedback) in an encrypted form. The general setup is as follows. First, state measurements are encrypted at the sensor. Associated ciphertexts are then transmitted to the cloud, where an encrypted control action is computed and sent to the actuator. Upon reception, the computed input is decrypted and applied to the system. Since data is never decrypted during transmission or in the cloud, eavesdropping and some other attacks are effectively prevented.

In this letter, we introduce an encrypted model predictive control (MPC) scheme for linear systems with state and input constraints. Thus, we extend the concepts from [3] and [4] to constrained systems, which is the main contribution of this letter. Technically, our approach exploits the fact that an MPC law is piecewise affine (for polytopic constraints and a quadratic performance criterion, see [5]). Under the assumption that this control law can be explicitly computed, we show that a secure and private cloud-based implementation of the controller can be realized using HE. In other words, we present an encrypted explicit MPC scheme.

In principle, encryption should not influence the performance of a controlled system since it only involves a temporary transformation of data. It turns out, however, that encryption requires quantization and it is well-known that quantization may affect the stability of controlled systems [6]. Thus, it is important to carefully address the effect of encryption on the system’s stability (similar to [4]). We here use robust MPC (as in [7]) to compensate quantization errors and to guarantee robust stability. While alternative methods for handling quantization exist (see [8]), we will show that the proposed robust MPC scheme additionally allows one to choose a suitable resolution and range of the quantization (i.e., a suitable fixed-point format). In this way, we provide another extension to [3] and [4].

This letter is organized as follows. We state notation in the remainder of this section and provide background on encryption and quantization in Section II. We then discuss the MPC design with quantization compensation in Section III. Our main result, i.e., the encrypted cloud-based evaluation of the resulting control law, is presented in Section IV. Finally, we illustrate the proposed control scheme with an example in Section V and state conclusions in Section VI.

Notation: We denote the sets of real, rational, integer, and natural numbers by $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively. We further introduce $\mathbb{N}_{[i,j]} := \{ j \in \mathbb{N} | i \leq j \}$ and $\mathbb{Z}_M$ as the set of integers modulo $M$ for $M \in \mathbb{N}$. For two sets $C, T \subseteq \mathbb{R}^n$, the operations $C \oplus T$ and $C \ominus T$ refer to the Minkowski addition and the Pontryagin difference, respectively. We frequently use the matrix norms $\|K\|_1$, $\|K\|_\infty$, and $\|K\|_{max} := \max_{j \in [i,n]} \max_{k \in [i,n]} |K_{jk}|$ for some matrix $K \in \mathbb{R}^{m \times n}$. The squared vector norm $\|\xi\|_P$ is understood as $\xi^TP\xi$ for some $\xi \in \mathbb{R}^n$ and some positive definite matrix $P \in \mathbb{R}^{n \times n}$.

II. BACKGROUND ON ENCRYPTION AND QUANTIZATION

A. Paillier Cryptosystem

In this letter, we use the Paillier cryptosystem (PCS, see [9]) to encrypt data. As most other cryptosystems, the PCS acts...
on a subset of the set of integers. The cardinality of this subset depends on the chosen key for the encryption. The key generation requires the selection of two large prime numbers $p_1$ and $p_2$ such that $\gcd(p_1p_2, (p_1 - 1)(p_2 - 1)) = 1$, where $\gcd(i, j)$ refers to the greatest common divisor of $i$ and $j$. For the simplified PCS used here, $p_1$ and $p_2$ are additionally required to be of the same order, i.e., $p_1, p_2 \in [2^{l-1}, 2^l]$ for some $j \in \mathbb{N}$. The public key then is $M = p_1p_2$. The private key (which is only known to the entity that needs to decrypt the data) is computed as $\lambda := \text{lcm}(p_1 - 1, p_2 - 1)$, where $\text{lcm}(i, j)$ refers to the least common multiple of $i$ and $j$. For the encryption of any plaintext $t \in \mathbb{Z}_M$, we now choose a random $r \in \mathbb{Z}_M^*$ and we construct the ciphertext as

$$E_M(t, r) := (M + 1)^rM \mod M^2 \in \mathbb{Z}_M^2.$$ 

For any ciphertext $c \in \mathbb{Z}_M^2$, the corresponding plaintext results from the decryption

$$D_M(c) := L_M(c^\lambda \mod M^2) \mu \mod M,$$

where $L_M(y) := (y - 1)\backslash M$ (with “\” denoting integer division) and where $\mu \equiv \lambda^{-1} \mod M$ is the so-called modular multiplicative inverse. It can be shown that encryption followed by decryption provides an integer equivalent to the plaintext. More formally, for every $t \in \mathbb{Z}_M$, we have $D_M(E_M(t, r)) = t$ for all $r \in \mathbb{Z}_M^*$. The PCS is a (partial) HE since it allows the computations of sums in encrypted form. More precisely, for every $t_1, t_2 \in \mathbb{Z}_M$ such that $t_1 + t_2 \in \mathbb{Z}_M$, we have

$$E_M(t_1, t_2)E_M(t_2, r) \mod M^2 = E_M(t_1 + t_2, r^2) \mod M^2$$

for all $r \in \mathbb{Z}_M^*$ [9, Sec. 8]. As a consequence of this property, we can also compute a semi-encrypted product. In fact, for every $t_1, t_2 \in \mathbb{Z}_M$ such that $t_1 t_2 \in \mathbb{Z}_M$, we obtain

$$E_M(t_1, t_2)^2 \mod M^2 = E_M(t_1 t_2, r^2) \mod M^2$$

for all $r \in \mathbb{Z}_M^*$. Both properties will be used to evaluate encrypted MPC laws in the remainder of this letter.

**B. Quantization and Fixed Point Arithmetic**

As detailed further below, the proposed application of the PCS requires quantization of the state measurements and the controller matrices. To this end, we consider signed fixed-point numbers in base 2 of the form

$$\beta_0 \beta_1 \beta_{l-1} \beta_{l-2} \cdots \beta_{d+1} \cdot \beta_d \beta_d-1 \cdots \beta_1,$$

where $l$ and $d$ denote the numbers of total and fractional bits, respectively. Based on this setup, all numbers in the set

$$\mathbb{Q}_{l,d} := \left\{ q \in \mathbb{Q} \mid \exists \beta \in \{0, 1\}^l : q = -2^{l-d-1} \beta_0 + \sum_{i=1}^{l-1} 2^{i-d-1} \beta_i \right\},$$

can be described. Obviously, $\mathbb{Q}_{l,d}$ contains rational numbers between $-2^{l-d-1}$ and $2^{l-d-1} - 2^{-d}$ separated from each other by the resolution $2^{-d}$. Now, the system states and controller matrices will, in general, be real-valued. We describe their quantization using the (surjective) mapping $q_{l,d} : \mathbb{R} \rightarrow \mathbb{Q}_{l,d}$ with

$$q_{l,d}(a) := \arg \min_{q \in \mathbb{Q}_{l,d}} |a - q|,$$

where the smallest absolute $q$ is selected in case two optimizers exist. To prepare the application of the PCS, we further introduce the bijective mapping $f_{l,d} : \mathbb{Q}_{l,d} \rightarrow \mathbb{Z}_{2^l}$ as

$$f_{l,d}(q) := 2^d q \mod 2^l.$$

Data will later be encrypted from and decrypted to the integer set $\mathbb{Z}_{2^l}$. However, the computed control actions will be values in $\mathbb{Q}_{l,d}$. We shall use the inverse $f_{l,d}^{-1} : \mathbb{Z}_{2^l} \rightarrow \mathbb{Q}_{l,d}$ with

$$f_{l,d}^{-1}(t) := \frac{1}{2^d} \left\{ \begin{array}{ll} t - 2^l & \text{if } t \geq 2^l - 1, \\ t & \text{otherwise} \end{array} \right.$$ 

To reconstruct the inputs. See [4, Prop. 1] for a proof that $q = f_{l,d}^{-1}(f_{l,d}(q))$ for every $q \in \mathbb{Q}_{l,d}$. Within the control law evaluation, we will later have to compute sums $q_1 + q_2$ and products $q_1q_2$ with $q_1, q_2 \in \mathbb{Q}_{l,d}$. Clearly, the sums and products will also be rational numbers. However, they might not be contained in $\mathbb{Q}_{l,d}$. Nevertheless, it is easy to see that there exists an $l_2 \geq l_1$ such that $q_1 + q_2$ and $2^d q_1 q_2$ are contained in $\mathbb{Q}_{l_2,d} \supseteq \mathbb{Q}_{l_1,d}$. We then find

$$f_{l_2,d}(q_1 + q_2) = 2^d (q_1 + q_2) \mod 2^{l_2} = (2^d q_1 \mod 2^{l_2} + 2^d q_2 \mod 2^{l_2}) \mod 2^{l_2} = (t_1 + t_2) \mod 2^{l_2},$$

where $t_1 := f_{l_1,d}(q_1)$ and $t_2 := f_{l_1,d}(q_2)$ (see [4, Prop. 2]). In other words, a mapped summation can roughly be interpreted as a sum in $\mathbb{Z}_{2^l}$. A similar observation holds for the product (see [4, Prop. 4]) in terms of

$$f_{l_2,d}(2^d q_1 q_2) = 2^d (2^d q_1 q_2) \mod 2^{l_2} = t_1 t_2 \mod 2^{l_2}.$$ 

**III. MPC DESIGN WITH QUANTIZATION COMPENSATION**

Our goal is to control linear systems of the form

$$x(k + 1) = A x(k) + B u(k), \quad x(0) = x_0$$

with polytopic state and input constraints

$$x(k) \in \mathcal{X} \quad \text{and} \quad u(k) \in \mathcal{U} \quad \text{for every } k \in \mathbb{N}.$$ 

Throughout this letter, we assume that the pair $(A, B)$ (with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$) is stabilizable and that the set $\mathcal{X} \subseteq \mathbb{R}^n$ is a convex and compact polytope with the origin as an interior point. We further assume that $\mathcal{U} \subseteq \mathbb{R}^m$ is a hyperrectangle, i.e.,

$$\mathcal{U} = \left\{ u \in \mathbb{R}^m \mid |u|_\infty \leq \overline{u}, \forall |j| \in \mathbb{N}_{[1,m]} \right\}$$

for some $\overline{u} \in \mathbb{R}^m$ with positive entries. We will show that the quantizations of the state measurements and controller matrices can be understood as artificial disturbances to the system (7) (see [10, Sec. IV] for a similar concept). We counteract those disturbances by applying a robust MPC scheme as in [7]. For the moment, we therefore consider the modified system

$$x(k + 1) = A x(k) + B u(k) + w(k), \quad x(0) = x_0$$

with additive disturbances from the set

$$\mathcal{W} := \{ w \in \mathbb{R}^n \mid |w|_\infty \leq \overline{w} \},$$

where $\overline{w} > 0$ can be understood as a design parameter. Roughly speaking, larger $\overline{w}$ allow larger quantization effects to be compensated. It turns out, however, that $\overline{w}$ cannot be chosen arbitrarily large. In fact, we have to guarantee that a matrix
$K^* \in \mathbb{R}^{m \times n}$ and a nonempty set $\mathcal{R}$ exist such that $A + BK^*$ is Schur and such that

$$\tag{12} (A + BK^*) \mathcal{R} \oplus \mathcal{W} \subseteq \mathcal{R}.$$ 

Based on such a robust positively invariant (RPI) set $\mathcal{R}$ (see [11]), we design the robust MPC scheme according to [7] and obtain the optimal control problem

$$V(x) = \min_{\hat{x}(0),\ldots,\hat{x}(N-1)} \|\hat{x}(N)\|^2_F + \sum_{k=0}^{N-1} \|\hat{x}(k)\|^2_Q + \|\hat{u}(k)\|^2_R$$

\[ \text{s.t. } x - \hat{x}(0) \in \mathcal{R}, \]
\[ \hat{x}(k+1) = A \hat{x}(k) + B \hat{u}(k), \quad \forall k \in \mathbb{N}_{[0,N-1]} \]
\[ \hat{x}(k) \in \mathcal{X} \subseteq \mathcal{X}, \quad \forall k \in \mathbb{N}_{[0,N-1]} \]
\[ \hat{u}(k) \in \mathcal{U} \cap K^* \mathcal{R}, \quad \forall k \in \mathbb{N}_{[0,N-1]} \]
\[ \hat{x}(N) \in \mathcal{T} \] \tag{13}

that is solved in every time step for the current state $x$ (i.e., $x(k)$). As in classical MPC, $P$, $Q$, and $R$ denote weighting matrices and $\mathcal{T}$ is a terminal set. In contrast to classical MPC, we consider tightened state and input constraints and we allow the initial state $\hat{x}(0)$ of the predicted state trajectory to differ from $x$. Based on the optimized state and input sequences $\hat{x}(0), \ldots, \hat{x}(N)$ and $\hat{u}(0), \ldots, \hat{u}(N-1)$, we apply the control law $\rho : \mathcal{F} \rightarrow \mathcal{U}$ with

$$\rho(x) := \hat{u}(0) + K^*(x - \hat{x}(0)),$$ \tag{14}

where $\mathcal{F} \subseteq \mathcal{X}$ is the set of all $x$ such that (13) is feasible. Under the assumption that $\mathcal{Q}$ and $\mathcal{R}$ are positive semidefinite and for a sensible choice of $P$ and $\mathcal{T}$ (see [7, Sec. 3] or Section V below for details), we can guarantee that the system

$$x(k+1) = A x(k) + B \rho(x(k)) + w(k), \quad x(0) = x_0 \tag{15}$$

is robustly stable for all $w(k) \in \mathcal{W}$ in the sense that the system will stay in $\mathcal{F}$ and converge to $\mathcal{R}$ for all initial states $x_0 \in \mathcal{F}$ [7, Th. 1].

Now, in case $\mathcal{R}$ and $\mathcal{T}$ are chosen as polytopes (which can be done according to [11] and [12]), it is well-known that (13) can be rewritten as a strictly convex quadratic program (QP) of the form

$$V(x) = \min_{Z} \frac{1}{2} Z^T H Z \quad \text{s.t. } G Z \leq F + x + d \tag{16}$$

with $Z^T = (\bar{Z}^T(0) \bar{u}(0) \ldots \bar{u}(N-1))$. As a consequence, the optimal decision variable $Z^* \in \mathbb{R}^{n+m}$ is a piecewise affine function of the current state $x$ (see [5]). This observation implies that the control law (14) is also piecewise affine. More precisely, (14) can be written as

$$\rho(x) = \begin{cases} K_1 x + b_1 & \text{if } x \in \mathcal{P}_1, \\ \vdots \\ K_s x + b_s & \text{if } x \in \mathcal{P}_s, \end{cases}$$ \tag{17}

where the sets $\mathcal{P}_1, \ldots, \mathcal{P}_s$ are convex polytopes with pairwise disjoint interiors and where $\bigcup_{i=1}^s \mathcal{P}_i = \mathcal{F}$. For systems with many states or for MPC schemes with long prediction horizons $N$, it is usually not possible to explicitly identify the structure (17) (with a reasonable numerical effort). However, for small systems with short prediction horizons, the controller matrices $K_i$ and $b_i$ as well as the polytopes $\mathcal{P}_i$ can be computed within seconds. Here, we assume that (17) can be computed offline and exploited online.

We will later use properties (1) and (2) to realize an encrypted evaluation of the affine control laws $K_i x + b_i$. Clearly, in order to apply the encryption $E_M$, we need to map the controller matrices and the current state to the set $\mathbb{Z}_M$. As a preparation, we consider the approximated control law

$$\rho(x) = \begin{cases} \hat{K}_1 \hat{x} + \hat{b}_1 & \text{if } x \in \mathcal{P}_1, \\ \vdots \\ \hat{K}_s \hat{x} + \hat{b}_s & \text{if } x \in \mathcal{P}_s, \end{cases}$$ \tag{18}

where $\hat{K}_i$, $\hat{b}_i$ and $\hat{x}$ result from the element-wise quantization of $K_i$ and $b_i$ (offline) and $x$ (online) via $g_{h_i,d}$. The quantized values can later be mapped to $\mathbb{Z}_M$ using (4). Two features of (18) are important. First, the quantization may lead to the problem that $\hat{u} := \rho(x) \notin \mathcal{U}$ for some $x \in \mathcal{F}$. In order to guarantee the required input constraint satisfaction, we will additionally apply the saturation function $\sigma : \mathbb{R}^m \rightarrow \mathcal{U}$ with

$$\sigma(\hat{u}) := \arg\min_{u \in \mathcal{U}} \| \hat{u} - u \|^2_2.$$ \tag{19}

Second, as detailed in Section IV, the selection of the controller segment $i$ in (18) is carried out based on the original state $x$ (and not $\hat{x}$). We next focus on the application of the approximated and saturated control law to the original system (7). Thus, we study the dynamics of the system

$$x(k+1) = A x(k) + B \rho((\hat{x}(k))), \quad x(0) = x_0.$$ \tag{20}

The following theorem states that, for a sensible design of the quantization in terms of $l_1$ and $d$, the dynamics (20) can be interpreted as a special instance of the disturbed system (15).

**Theorem 1:** Let $x_0 \in \mathcal{F}$ and let $l_1$ and $d$ be such that

$$\max\{\|K_i\|_{\infty}, \|b_i\|_{\infty}, \|x\|_{\infty}\} \leq \frac{1}{2^{l_1-d-1} - 2^{-d-1}},$$

$$\|K_i\|_{1} + \|x\|_{1} + n 2^{-d-1} + 1 \leq \sigma\|B\|_{\infty} 2^{l_1-d+1}$$

for every $i \in \mathbb{N}_{[1,s]}$ and every $x \in \mathcal{F}$. Then, there exists a sequence $w(0), w(1), \ldots \in \mathcal{W}$ such that the state trajectories of the systems (15) and (20) are equal.

**Proof:** We initially show that

$$\|B\sigma(\rho(x) - \rho(x))\|_{\infty} \leq \bar{w},$$ \tag{23}

for every $x \in \mathcal{F}$. To this end, we overestimate the right-hand side in (23) by $\|B\|_{\infty} \sigma(\rho(x) - \rho(x))$. Now, since $\rho(x) \in \mathcal{U}$ holds by construction, we have

$$\|\sigma(\rho(x) - \rho(x))\|_{\infty} \leq \|\rho(x) - \rho(x)\|_{\infty}$$

according to Lemma 1 in the Appendix. As a consequence, (23) holds if

$$\|\rho(x) - \rho(x)\|_{\infty} \leq \|B\|_{\infty} \sigma.$$ \tag{24}

Without loss of generality, we assume that $x$ is contained in $\mathcal{P}_i$ and that $\hat{x}$ is the quantization of $x$ based on $g_{h_i,d}$. Thus,

$$\|\hat{x} - x\|_{\infty} = \|\hat{K}_i \hat{x} + \hat{b}_i - K_i x - b_i\|_{\infty}.$$ \tag{25}

Clearly, (21) implies that

$$\max\{\|\Delta K_i\|_{\infty}, \|\Delta b_i\|_{\infty}, \|\Delta x\|_{\infty}\} \leq 2^{-d-1},$$

where $\Delta K_i := \hat{K}_i - K_i$, $\Delta b_i := \hat{b}_i - b_i$, and $\Delta x := \hat{x} - x$. Hence, we find

$$\|\hat{K}_i \hat{x} + \hat{b}_i - K_i x - b_i\|_{\infty} \leq \|\Delta K_i x + K_i \Delta x + \Delta K_i \Delta x + \Delta b_i\|_{\infty} \leq \|K_i x\|_{\infty} + \|\Delta K_i\|_{\infty} + \|\Delta x\|_{\infty} + \|\Delta b_i\|_{\infty} \leq \|x\|_{1} 2^{-d-1} + \|K_i\|_{1} 2^{-d-1} + n (2^{-d-1})^2 + 2^{-d-1}.$$ \tag{26}

Combining (25) and (26) with condition (22) yields (24).
IV. ENCRYPTED CONTROL LAW EVALUATION

The implementation of the quantized predictive controller requires to evaluate \( \sigma(\hat{\rho}(x)) \) for the current state \( x \) in every time step. After having identified the polytope \( i \) such that \( x \in \mathcal{P}_i \), the control action results from computing

\[
u = \sigma(\hat{u}) \quad \text{with} \quad \hat{u} := \hat{\rho}(x) = \hat{R}_i \hat{x} + \hat{b}_i. \tag{27}
\]

We state the following theorem to prepare a secured computation of \( \hat{u} \) based on encrypted data.

**Theorem 2:** Let \( x \in \mathcal{P}_i \), let conditions (21) and (22) be satisfied, and let \( I_2 \) and \( M \) be such that

\[
(2^{l_i-1} - 1)(n(2^{l_i-1} - 1) + 2^d) + 1 \leq 2^{l_i-1} \quad \text{and} \quad 2\hat{l}_i + n(2^{l_i-1} - 1)^2 \leq M. \tag{28}
\]

Then, the \( j \)-th component of \( \hat{u} \) satisfies

\[
\hat{u}_j = 2^{-d}f_{I_2,d}^{-1}(D_M\left(\prod_{k=1}^{n} c_k^j \mod M^2\right) \mod 2^{l_i}), \tag{30}
\]

where

\[
c_k := E_M(f_{I_2,d}(\hat{x}_k), r_k), \quad \forall k \in \mathbb{N}_{[1,n]}, \tag{31}\]

\[
t_k := f_{I_2,d}((\hat{R}_i)k), \quad \forall k \in \mathbb{N}_{[1,n]}, \tag{32}\]

\[
c_0 := E_M(f_{I_2,d}(2d(\hat{b}_i)), r_0) \tag{33}
\]

for arbitrary \( r_0, \ldots, r_n \in \mathbb{Z}_M^*.

**Proof:** We obviously have

\[
\hat{u}_j = 2^{-d}f_{I_2,d}^{-1}(f_{I_2,d}(2^d(\hat{R}_i \hat{x} + \hat{b}_i))) \tag{34}
\]

under the assumption that

\[
2^d(\hat{R}_i \hat{x} + \hat{b}_i) \in \mathbb{Q}_{I_2,d} \tag{35}
\]

Since \( \hat{R}_i, \hat{b}_i, \hat{x} \in \mathbb{Q}_{I_2,d} \), it is easy to see that \( 2^d(\hat{R}_i \hat{x} + \hat{b}_i) \) is an integer multiple of \( 2^{-d} \). Thus, (35) holds if

\[
\left|2^d(\hat{R}_i \hat{x} + \hat{b}_i)\right| \leq 2^{l_i-d-1} - 2^{-d}. \tag{36}
\]

Clearly, (21) implies

\[
\left|2^d(\hat{b}_i)\right| \leq 2^d\left\|\hat{b}\right\|_{\infty} \leq 2^d(2^{l_i-d-1} - 2^{-d}) = 2^{l_i-1} - 1
\]

and

\[
\left|2^d(\hat{R}_i \hat{x})\right| \leq 2^dn\left(2^{l_i-d-1} - 2^{-d}\right)^2 \leq n2^{-d}\left(2^{l_i-1} - 1\right)^2.
\]

The two latter relations in combination with (28) immediately prove (35). It remains to show that

\[
f_{I_2,d}(2d(\hat{R}_i \hat{x} + \hat{b}_i)) = D_M(v_j) \mod 2^{l_i},
\]

where

\[
v_j := c_0 \prod_{k=1}^{n} c_k^j \mod M^2 = c_0 \prod_{k=1}^{n} (c_k^j \mod M^2) \mod M^2. \tag{36}
\]

Relation (36) can be rewritten using properties (1) and (2). In fact, under the assumption that \( t_k f_{I_2,d}(\hat{x}_k) \in \mathbb{Z}_M \), we have

\[
c_k^j \mod M^2 = E_M(t_k f_{I_2,d}(\hat{x}_k), r_k^j) \tag{37}
\]

according to (2). Taking (1) into account and defining

\[
t_{res} := f_{I_2,d}(2d(\hat{b}_i)) + \sum_{k=1}^{n} t_k f_{I_2,d}(\hat{x}_k) \quad \text{and} \quad r_{res} := r_0 \prod_{k=1}^{n} r_k^j,
\]

we further obtain \( v_j = E_M(t_{res}, r_{res}) \) under the assumption that \( t_{res} \in \mathbb{Z}_M \). Since \( f_{I_2,d} \) is mapping to \( \mathbb{Z}_M \), both assumptions are satisfied due to (29). We consequently find

\[
D_M(v_j) = t_{res} \quad \text{and finally}
\]

\[
t_{res} \mod 2^{l_i} = f_{I_2,d}(2d(\hat{b}_i)) + \sum_{k=1}^{n} t_k f_{I_2,d}(2d(\hat{R}_i)j_k \hat{x}_k) \mod 2^{l_i}
\]

\[
= f_{I_2,d}(2d(\hat{b}_i)) + \sum_{k=1}^{n} 2^d(\hat{R}_i)j_k \hat{x}_k
\]

\[
= f_{I_2,d}(2d(\hat{R}_i \hat{x} + \hat{b}_i)).
\]

according to (5) and (6), which completes the proof.

**Theorem 2** is central for our approach. Most importantly, the right-hand side of (30) enables an encrypted cloud-based evaluation of the controller. The required computations at the sensor, in the cloud, and at the actuator are specified in the following and the resulting control scheme is illustrated in Figure 1. In this context, a proper operation of the controller is guaranteed by conditions (28) and (29) that regulate the mapping \( f_{I_2,d} \) and the encryption \( E_M \).

A. Computations at the Sensor

At the sensor, we measure the current state \( x \) (i.e., \( x(k) \)) at every time step and we identify the polytope \( \mathcal{P}_i \) such that \( x \in \mathcal{P}_i \). In this context, binary search trees can be used for
an efficient and easy to implement identification of the correct set $\mathcal{P}_i$ [13]. We then quantize $x$ by applying the mapping $g_{1,d}$ element-wise. Afterwards, the quantization $\tilde{x}$ is element-wise encrypted by evaluating $c_k$ according to (31). The resulting ciphertexts $c_1, \ldots, c_n \in \mathbb{Z}_M^2$ are then transmitted to the cloud. More precisely, corresponding to the identified region $\mathcal{P}_i$, we call the program $i$ with the arguments $c_1, \ldots, c_n$.

\section{Computation in the Cloud}

In the cloud, $s$ programs are uploaded offline corresponding to the $s$ polytopic regions $\mathcal{P}_i$. Every program $i$ computes an encrypted version of the quantized inputs $\tilde{u}_1, \ldots, \tilde{u}_m$ by partially evaluating (30). Let the function $j$ in program $i$ be responsible for evaluating $\tilde{u}_j$ in $\mathcal{P}_i$. Each function $j$ is equipped with $n$ hard-coded parameters $\theta_i$ as in (32) and the hard-coded ciphertext $c_0$ as in (33). In every time step, corresponding to the polytope $\mathcal{P}_i$ identified at the sensor, the program $i$ is called with the encrypted state information $c_1, \ldots, c_n$. Within the program, every function $j \in \mathbb{N}_{1, m}$ is called and the output $v_j$ as in (36) is computed. Obviously, $v_j \in \mathbb{Z}_M^2$ is related to the encrypted input $\tilde{u}_j \in \mathbb{Q}_{0, d}^2$. This can easily be seen from the decomposition (30). Finally, the ciphertexts $v_1, \ldots, v_m$ are transmitted to the actuator.

\section{Computation at the Actuator}

At the actuator, it remains to decrypt, transform, and rescale the ciphertexts $v_1, \ldots, v_m$ to obtain $\tilde{u}_1, \ldots, \tilde{u}_m$. According to (30), these computations are done by evaluating

$$\tilde{u}_j = 2^{-d-d_j^{-1}} (D_M(v_j)) \mod 2^d$$

for every $j \in \mathbb{N}_{1, m}$. Finally, we have to guarantee input constraint satisfaction by evaluating $u = \sigma(\tilde{u})$ as in (19). For the considered hyperrectangular set $U$, this can be done according to (39) in the Appendix. The resulting control action is then applied to the system.

\section{Numerical Example}

We illustrate the proposed encrypted MPC scheme with a double integrator. The system matrices read

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

and the constraints are $X = \{x \in \mathbb{R}^2 \mid |x_1| \leq 25, |x_2| \leq 5\}$ and $U = \{u \in \mathbb{R} \mid |u| \leq 1\}$. The weighting matrices $Q = I$ and $R = 0.01$, the set $W$ with $\sigma = 0.1$, and the prediction horizon $N = 9$ are chosen as in [7]. We stress again that $\sigma$ is a design parameter that influences the quantization in terms of condition (22). Next, the terminal weighting matrix $P$ is selected as the solution of the Riccati equation

$$A^T(P - PB(R + B^TPB)^{-1}B^TP)A - P + Q = 0$$

and the stabilizing controller gain is chosen as

$$K^* = -(R + B^TPB)^{-1}B^TPA \approx (-0.6609 \quad -1.3261).$$

We then use the method from [11] to compute an RPI set $\mathcal{R}$ satisfying condition (12) and obtain

$$\mathcal{R} = \text{conv}\left\{ (0.2536, 0.0516, 0.2536, 0.0499) \right\}.$$

Finally, the terminal set $T$ is chosen as

$$T := \{ x \in \mathbb{R}^d \mid (A + BK^*)^T x \in C, \forall k \in \mathbb{N} \},$$

where $C := \{ x \in X \otimes R \mid |Kx| \in U \cup K^* R \}$. Note that $C$ is polytopic since $X, U, \text{and} R$ are polytopes. As a consequence, $T$ is polytopic as apparent from [12, Th. 4.1]. We now formulate the robust MPC scheme (13) as aQP of the form (16) and solve it explicitly according to [5] using the multiparametric toolbox [14].

This letter focuses on the encrypted evaluation of the explicit MPC law within a cloud. We consequently provide a detailed implementation of the procedure from Section IV that allows to reproduce our numerical results. We first design the quantization $g_{1,d}$ such that the conditions (21) and (22) are satisfied. As a preparation, we compute

$$\max_{i \in \mathbb{N}_{1, d}} ||K_i||_{\max} \approx 7.2737, \quad \max_{i \in \mathbb{N}_{1, d}} ||K_i||_1 \approx 7.9346,$$

$$\max_{x \in F} ||x||_1 = 25, \quad \max_{x \in F} ||x||_1 = 30,$$

$$\max_{i \in \mathbb{N}_{1, d}} ||b_i||_{\max} \approx 27.1896.$$

We then choose the smallest $d \in \mathbb{N}$ such that condition (22) holds (with $n = 2$) and obtain $d = 8$. Next, we identify the smallest $l_1 \in \mathbb{N}$ such that (21) is satisfied and find $l_1 = 14$. Afterwards, we design the mapping $f_{1, d}$ to $Z_{l_1, z_2}$. In this context, the smallest $l_2 \in \mathbb{N}$ satisfying (28) is $l_2 = 29$. To prepare the encryption, it remains to choose suitable primes $p_1$ and $p_2$ such that $M = p_1 p_2$ satisfies (29). However, not only this inequality is important for the choice of $p_1$ and $p_2$. In fact, the encryption should also be safe which requires $p_1$ and $p_2$ to be very large (see [9, Sec. 6]). Here, for illustration purposes, we choose the relatively small primes $p_1 = 3320791807$ and $p_2 = 4075816727$ in $[2^{31}, 2^{32}]$. After computing the public key $M = p_1 p_2 = 13534938798355156569$ and the private key $\lambda = \text{lcm}(p_1 - 1, p_2 - 1) = 13534938768458547156$, we are ready to apply our encrypted MPC scheme.

To illustrate the application, we consider the initial state

$$x_0 = (-2.28125 - 2^{-d-1} 3.125 + 2^{-d-1})^T. \quad (37)$$

At time step $k = 0$, the sensor measures the current state $x = x(0) = x_0$ and identifies the polytope $\mathcal{P}_i$ containing $x$. Afterwards, the state $x$ is quantized using $g_{1,d}$ and we obtain $\tilde{x} = (-2.28125 \ 3.125)^T$. Next, the mapping via $f_{1, d}$ and the encryption with the random numbers $r_1 = 743769631872510247$ and $r_2 = 3862701628878047690$ results in the ciphertexts

$$c_1 = 940996016497879821937728911142536192,$$

$$c_2 = 25999183088754191992843264937885822755,$$

as in (31) that are transmitted to the cloud.

In the cloud, the program $i$ (corresponding to the polytope $\mathcal{P}_i$) is called. In plaintext, it implements a quantized version of the locally affine control law

$$K_i x + b_i \approx (-0.0996196 \quad -0.7586155)x + 1.0699914.$$
as in (33) for \( r_0 = 6170687951983498748 \). These numbers are hardcoded in the function \( j = 1 \) within program \( i \). For this example, due to \( m = 1 \), each of the \( s \) programs in the cloud contains only one function. In case \( m > 1 \), the previous steps are repeated for every component \( j \) of the control law. Now, online, we compute \( v_1 \) according to (36), obtain

\[
v_1 = 2868333910292456070456110032988022180,
\]

and transmit it to the actuator.

At the actuator, we first evaluate

\[
\hat{u} = 2^{-d f_{2,d}} \left( D_M(v_1) \mod 2^d \right) = 1.01904296875.
\]

Clearly, \( \hat{u} \) violates the input constraints. However, after evaluating the saturation function, we obtain \( u = \sigma(\hat{u}) = 1 \in \mathcal{U} \), which is finally applied to the system. The procedure is then repeated for every step \( k \in \mathbb{N} \). The resulting trajectory is illustrated in Figure 2. As desired, the system is steered towards the origin or, more precisely, to the RPI set \( \mathcal{R} \).

VI. CONCLUSION

We presented an encrypted implementation of an MPC scheme for linear systems with polytopic constraints. The encrypted evaluation of the control law builds on the explicit computation of the piecewise affine control law combined with homomorphic properties of the Paillier cryptosystem. A proper operation of the encrypted controller is guaranteed by a set of conditions (namely (21), (22), (28), and (29)) that regulate the operation of the encrypted controller is guaranteed by a set of homomorphic properties of the Paillier cryptosystem. A proper encrypted evaluation of the control law builds on the explicit computation of the MPC law.

As discussed in Section III, this approach is only practical for systems with a few states and short prediction horizons. The extension to more complex systems therefore requires to focus on encrypted online optimization strategies similar to [15]. Clearly, this leads to a different setup. Nevertheless, we are confident that some features of the proposed scheme, e.g., the quantization compensation via robust MPC, are portable.

APPENDIX

Lemma 1: Let \( \mathcal{U} \) and \( \sigma \) be defined as in (9) and (19), respectively. Further, let \( \hat{u} \in \mathbb{R}^m \) and \( u \in \mathcal{U} \). Then

\[
\| \sigma(\hat{u}) - u \|_{\infty} \leq \| \hat{u} - u \|_{\infty}
\]

Proof: Due to the hyperrectangular shape of \( \mathcal{U} \), we have

\[
\sigma_j(\hat{u}) = \text{sign}(\hat{u}_j) \min \left\{ |\hat{u}_j|, \bar{u}_j \right\}
\]

for every \( j \in \mathbb{N}_{1,m} \). Taking into account that \( |\hat{u}_j| \leq \bar{u}_j \) due to \( u \in \mathcal{U} \), we easily prove that \( |\sigma_j(\hat{u}) - u_j| \leq |\hat{u}_j - u_j| \) for every \( j \in \mathbb{N}_{1,m} \), which immediately leads to (38).

REFERENCES


